REPORT DOCUMENTATION PAGE

Form Approved
OMB NO. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	14 May 02	Final, 05 Jul 00 – 31 Oct 01

4. TITLE AND SUBTITLE

Tomographic Imaging on Distributed Unattended Ground Sensor Arrays

5. FUNDING NUMBERS

ARO contract 41514-PH
DAAD19-00-C-0099

6. AUTHOR(S)

D. J. Brady, R. Morrison, D. C. Munson, Jr., R. Stack

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Beckman Institute and Coordinated Science Laboratory
University of Illinois
Urbana, IL 61801

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

U. S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision, unless so designated by other documentation.

12 a. DISTRIBUTION / AVAILABILITY ST ATEMENT

Approved for public release; distribution unlimited.

12 b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

The objective of this project was to demonstrate tomographic data fusion from distributed ground sensor arrays. The project leveraged other DARPA work by the same researchers at the University of Illinois on the topic of interferometric digital imaging. The current program was oriented toward design of an array of imaging sensors, including both cameras and interferometric sensors, to accurately and robustly locate and track ground targets.

Over the course of this project we designed and constructed five first-generation sensor modules. Four of these modules were fitted with 180-degree panorama video sensor heads, while the fifth used a sensor head consisting of a rotational shearing interferometer. Two modules were used in field tests with IR cameras replacing the visible-light sensor heads. In a second field test, four modules with visible-light CMOS cameras were used to demonstrate tomographic imaging of a human subject moving through a test environment. In addition, a smaller, low-power second generation module, based on the StrongArm processor, was designed, assembled and tested.

14. SUBJECT TERMS	15. NUMBER OF PAGES
	16. PRICE CODE

17. SECURITY CLASSIFICATION OR REPORT	18. SECURITY CLASSIFICATION ON THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	**UNCLASSIFIED**	**UNCLASSIFIED**	**UL**

Final Report - DAAC19-00-C-0099 "Tomographic Imaging on Distributed Unattended Ground Sensor Arrays"

David Munson - University of Illinois
Rick Morrison - Distant Focus Corporation

May 14, 2002

Contents

1 Objectives

The purpose of this project was to demonstrate tomographic data fusion from distributed ground sensor arrays. This project built upon basic sensor science technologies developed under support from the DARPA Defense Sciences Office. The goal under this program was to make an unattended sensor array that can accurately and robustly locate and tracks targets.

At the completion of this project there were five first generation sensor modules constructed. Four of these modules were fitted with 180-degree panorama video sensor heads while the fifth contained a sensor head consisting of a rotational shear interferometer (RSI). Two modules were used in field tests with two IR cameras replacing the visible sensor heads. In a second field test, four modules and visible spectrum CMOS cameras were used to demonstrate tomographic modeling of a human subject moving through a test environment. In addition, a second generation module based on the StrongArm processor was assembled and tested.

2 Sensing and Processing Modules

2.1 First Generation Sensor Module

During the first half of the project, five first generation sensor modules were constructed. The basic sensor module is shown in figure 1 and the internal processor core is shown in figure 2. Each core sensor module contained a Pentium-class processor operating at 266 MHz and a wireless 802.11b standard network card. Sensor modules were also fitted with a video capture card and a digital-to-analog converter card as needed. Sensor heads consisted of four CMOS cameras placed at 45° angles around a half-circle. A 10μm IR camera replaced a CMOS camera during certain tests. An interferometric sensor head in the form of an RSI, developed under another contract at the University of Illinois from the DARPA Applied Mathematics and Computation Program, was fitted to one module for testing. The sensor modules also interfaced to data acquisition and control workstations as well as visualization hardware for our test and evaluation purposes.

As expected, due to design and budget constraints, the first generation sensor modules used substantially more energy, weighed more, and transfered more information than was necessary or desirable. Nevertheless, these prototypes provided detailed and essential feedback for component and algorithm development. Embedding processing functions in the sensor and communication network is the key to reducing system resource requirements. Substantial improvements on the initial module design were clearly possible and were incorporated into our second-generation modules. In fact, these first generation prototypes allowed us to test and simulate a variety of embedded processing approaches, to create a

Figure 1: Completed sensor module

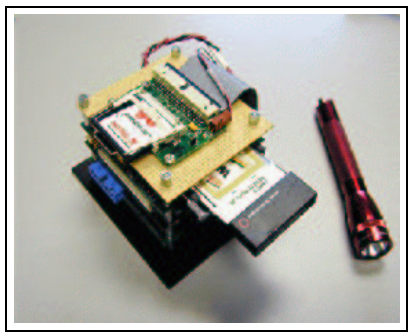


Figure 2: Sensor module processing core

Figure 3: Panoramic video display obtained from sensor module

basis for analyzing energy, bandwidth and time budgets, and to refine designs for second generation modules.

In early November 2000, an operating sensor array of three video modules and one specialized RSI module were demonstrated at the SPIE Law Enforcement Technologies conference in Boston. During this presentation on the unattended ground sensor array program, real-time operation of the wireless video system was demonstrated to the audience. For example, the 180-degree panoramic image (as shown in figure 3) created from the four camera sensor head was selected using the easy-to-use interface. In addition, a rudimentary tracking program, using two of the modules, was demonstrated.

2.2 Design of second generation modules

Second generations module design focused initially on selection of processing platform. The main considerations were power, speed, and ease of design. As will be explained in the following section, we performed benchmarking on the three potential contenders for the second generation modules. The three contenders were the PC104 platform using the Intel Pentium processor, the MachZ processor from ZF-Linux Corporation, and the Intel StrongARM processor. Originally we had targeted the MachZ solution for our second generation modules, as the first generation PC104 platforms consumed considerable power and were quite bulky. The MachZ is a single chip computing system solution. We eventually identified a better solution based on the Intel StrongARM processor, which forms the core of the Compaq iPAQ PDA. While a module consisting of a StrongARM processor would not be a single chip solution and would be potentially very challenging to design, a development board design complete with CAD files was available from Intel. This development design contained all functions that we had hoped to incorporate into our module design, including Lithium-ion battery support.

2.3 Benchmark of potential 2G module processors

One project goal was to evaluate and design a second generation unattended ground sensor module based on experience gained from the first generation module. Two critical features of the new module were the processing power available and the power consumption of the processor chip set. We examined two new chip sets, the MachZ from ZF-Linux Corporation and the Intel StrongARM 1110 chip set, and compared these to the original PC104 platform.

We defined and tested benchmarks that aided us in determining the proper choice of processor for the next generation of modules. Using the MachZ development kit and a Compaq iPAQ PDA containing the StrongARM 1110, we developed software that made rigorous use of the processor systems and targeted specific attributes of each. Each platform was adapted to run a version of the Linux operating system. Source code for the tests was developed on a local workstation and ported to the systems to be natively compiled. Benchmarking included a range of tests; from millions of additions, subtractions, multiplication, and divisions, to floating point arithmetic, and memory access. System specific tests included pseudo-random memory access. Each test was executed multiple times to insure repeatability. Tests also included software options such as compiler optimizations, which construct larger files and have longer compile times, but deliver better performance.

2.4 Some Considerations

Our studies showed that, in general, the Intel StrongARM processor would be the better choice. In addition to power saving capabilities that are world

Processor	StrongARM	MachZ	Pentium MMX
	206 MHz	133 MHz	266 MHz
	0.4 Watts	1.79 Watts	7.6 Watts
10M adds/subs	0.07	0.26	0.04
10M mults/divs	4.78	2.95	0.74
1M FP mults	10.2	0.53	0.07
Memory Access	1.29	1.73	1.03
Pseudo random access	0.72	1.62	0.52

Table 1: Processor comparison with compiler optimizations (all units defined in seconds). Smaller numbers are better performance.

class for processors of its performance, the StrongARM is an easy processor to integrate with a heterogeneous system. The StrongARM has performed better than the MachZ in a number of categories, specifically those which involve simple operations, like additions and subtractions. The StrongARM also provided faster memory access than the MachZ. However, the StrongARM suffered from the lack of a Floating Point Unit (FPU) and therefore suffered in the area of floating point arithmetic. However, this could be overcome by reducing floating point operations in code and by coding workarounds when necessary. For those operations that are inherent in the C language and in the operating system, the StrongARM Linux kernel provides a FPU software simulator workaround, but this was found to be slow when compared to hardware FPUs.

We were also encouraged by the availability of the schematics and PC board design files for the StrongARM platform. With only a little modification to these files, future custom modules could be constructed.

We located a vendor, ADS, that a sells StrongARM development platforms. One of their systems ran Linux and had a USB master feature. The drawback was that the package was a larger form factor than we desired. However, ADS had a second system, the Bitsy, that was the desired size although not outfitted with Linux. In order to meet the objectives of the project, we purchased the first development system from ADS in order to develop our algorithms on this StrongARM platform and then transitioned this system to the Bitsy development package as soon as it was available with Linux.

Our decision to use a StrongARM platform with a USB master feature and Linux operating system was based on the power conserving processor, the wide availability of USB cameras, and the open source nature of Linux that provided us with a conduit for rapidly developing and integrating code.

One of the important design decisions that we made related to the interface of the sensor to the processor. We chose to use USB since it is widely accepted, and one could plug multiple sensors into one processor. USB also has sufficient bandwidth for imaging sensors. USB offered the advantage of being able to use commercially available devices as well as custom-made sensors. It also made interchanging sensor heads, either for repair or for reconfiguration, trivial. Our

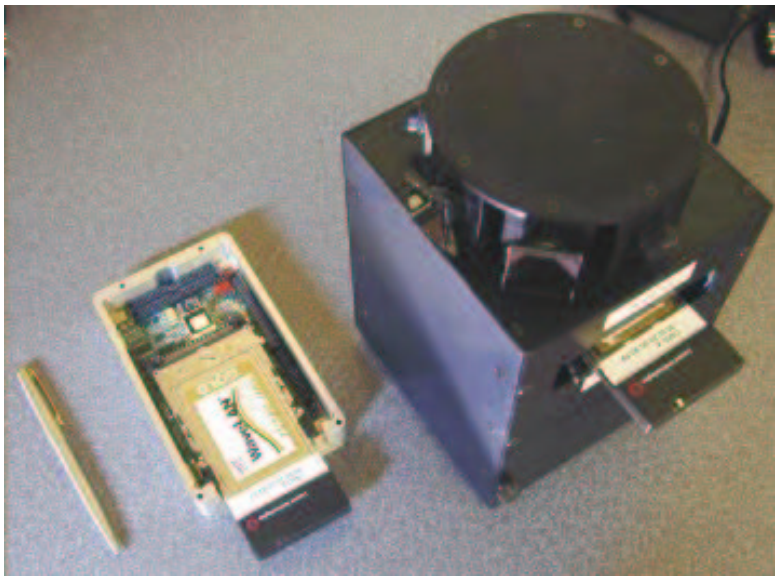


Figure 4: Second generation sensor module (left) next to first generation module.

first sensor was a USB camera made by Creative, it was a 640x480 color CMOS imager using the OV511 controller.

2.5 Second Generation Module Prototype

The machine shop manufactured the processor housing, the camera cover and battery housing. This part is shown in figure 4 next to the Pentium-based module constructed at the start of the contract. More details of the processor core and the battery compartment are shown in figure 5 and figure 6. We designed the module to accept several sizes of batteries, giving the unit a range of operating times from 4 hours to nearly 24 hours. This was longer than our original design goal of 12 hours. The size of the module even with a larger battery was still much smaller than the first generation module, measuring about 6"; wide by 3"; tall by 6"; deep (4"; with smaller battery). The module, if assembled properly, was waterproof, making outdoor use feasible in all weather conditions. One problem we encountered was with the battery charging components. The charging circuitry on the processor board was not functional; this was an ADS problem that they were working on. Also the board only handled a maximum of 12V. Thus we had to install an additional DC to DC converter to power the processor board and an external charger to recharge the battery as well as power the unit. The main problem with this was that we were not able to effectively monitor the battery life in software. ADS is currently resolving both issues,

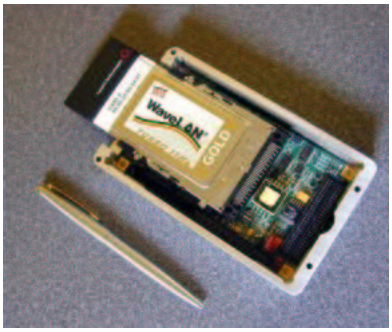


Figure 5: Second generation module processing core.

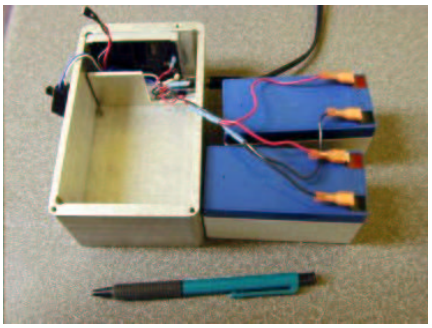


Figure 6: Second generation module battery compartment.

thus additional units will be more power efficient and will be able to monitor battery life.

The graphics master development system proved invaluable in the design and testing of the custom Linux distribution. The main focus was on debugging the USB interface on the StrongArm platform and writing software to allow image acquisition from the USB camera we used. Our main problem was the unstable USB interface under the StrongArm Linux operating system.

3 Tomographic Analysis and Code Development

3.1 Overview and basic image acquisition and control

Let us begin with a quick review of the basic software architecture of the sensor array system. The basic applications used in the network can be separated into two primary classes: (1) control and display applications that run on workstations, laptops, and personal digital assistants (PDA), and (2) data acquisition, analysis, and server applications that run on the sensor modules. Both classes communicate control information and data through socket connections over a wireless 802.11b network.

Class 1 applications are typically Java applets that can be run independently or within a web browser framework. Most of these applets are designed to be served to the remote display unit via the web server running on the sensor module. There are also some simple display interfaces for the PDAs. Although a small portion of the tomographic analysis is performed in these applications (such as the collection and synthesis of a tomographic model from each individual module's data), the primary function is data collection and display.

Class 2 applications are typically C coded programs installed in a Linux environment. C-coded applications assure the highest performance and access to lower level drivers for various hardware elements. The two primary classes are

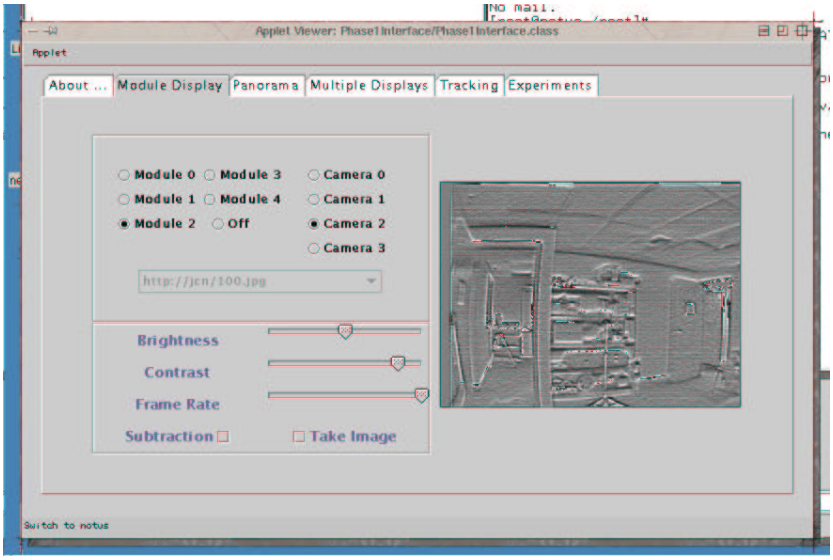


Figure 7: Illustration of video display with image processing filter

imageserver (for collecting, compressing, storing and streaming video data) and volumeserver (for creating module specific tomographic reconstructions.)

The sensor module's "*imageserver*" application (class 2) was evolved significantly over the course of the project. Imageserver is responsible for triggering the video frame capture and streaming the image to either a file or a socket connection. It also performs the module specific tomographic reconstruction elements. The following major enhancements were added:

- The *Lua* extension programming language was embedded as the command interpreter in *imageserver*. This provided a means for scripting and quickly adding new commands. In the process of incorporating *Lua*, commands and image manipulation routines were separated into a shared library.
- Image processing filters were incorporated in the video section. Filters included median and mean filters and a 3x3 kernel-based filter. A Laplacian filter is demonstrated in figure 7.
- Software applications were under change control supervision using CVS. This process is typical of large coding projects and provides benefits in tracking code development.

Perl scripts were also created for providing "video tape" functionality. These Perl scripts were used by the web server on the sensor modules. We also directly installed the command/display Java applets directly onto the modules so that

the sensor module's web server provided all primary software elements of sensor array operation.

The initial tomographic analysis was based on a simple triangulation method. An object was identified in the field of view of two sensor module cameras. The centroid of each object was then traced to a common intersection and the position was reported. The location was calculated for two dimensions in the first field test. Details of a field study based on this approach were reported in an attached document.

The advanced tomographic algorithm is based on the silhouette approach. The images acquired by each module undergo background subtraction and filtering in preparation for analysis. Our objective was to start with a low resolution reconstruction in order to achieve rates of between one and ten reconstructions per second. Four major software components were implemented for this tomographic application:

- geometry definition
- module specific volume modeling
- intermodule communication and control
- and model synthesis and display

In this tomographic application, geometry definition is the process of determining which pixels are associated with specific volume elements (voxels.) Location and orientation of the module, camera, and volume region are measured and submitted to the program. The application then calculates the database of correlations. The process can be scaled to various image sizes and voxel resolutions. We most often worked with 1/8 resolution monochrome images (40×30) and $16 \times 16 \times 16$ volumes so that the database could be constructed on the modules in a reasonable time (about one minute) and so that reconstruction occurred at near video rates.

During the volume reconstruction phase, each sensor module produced volumetric information for one camera view. The reconstruction was based on back projection of a background-subtracted image in the style of the silhouette algorithm.

Next, the control/interface program running on the laptop collected the models from each of the sensor modules. Each of the models was combined to form a single model. The implementation required the laptop to provide the control and act as the central link in collecting the models. A further enhancement would be to distribute this functionality among the various modules. In this situation, all modules would interact with each other such that each module is capable of providing the full volumetric model. In this situation, a personal digital assistant (PDA) could gather and present information from the sensor space.

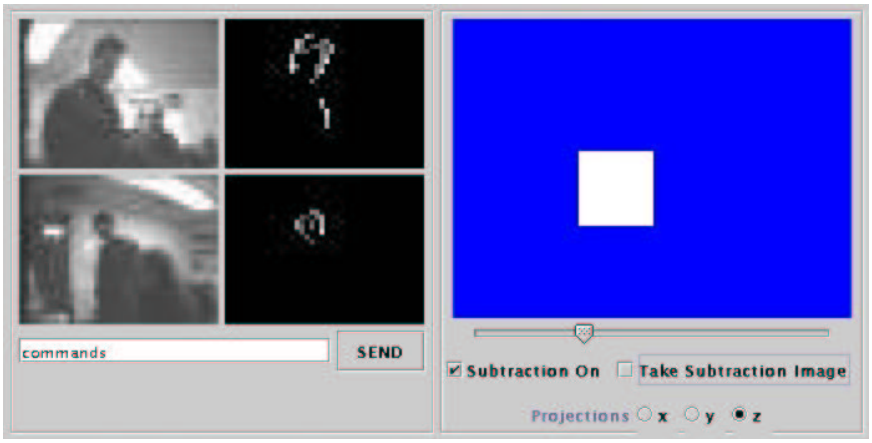


Figure 8: Early tomographic reconstruction using three modules. Two video streams and their background subtracted elements are on the left. The location of the subject within the volumetric reconstruction is shown on the right from above.

Shown in figure 8 is a display example from the tomographic reconstruction in action. This was an early implementation using only two cameras. By the completion of the project, three or more modules were used to generate the reconstruction. The images on the left-hand side of the display are raw video feeds from two modules. The two central images are background-subtracted images showing motion of the human subject. On the right-hand side of the figure is a projection, from above, of the volume reconstruction showing the subject's position in the volume.

Performance is highly dependent on whether or not a background image is acquired and stored between every frame analysis. Without background subtraction, the tomographic reconstruction using three sensor modules calculated and displayed about 3 reconstruction models/sec. This reconstruction model was a $16 \times 16 \times 16$ voxel volume created from an 1/8 resolution (40×30) downsampled image. With background subtraction and interframe background acquisition enabled, the three module rate fell to about 1.7 frames/sec.

Several aspects of the code were examined closely and modified. Eliminating the video display from the three cameras on the Java control interface module significantly increased performance, as expected. The basic background subtraction at this resolution had little impact on performance. Coordinating the three modules to process data synchronously and independently rather than in a sequential mode also aided performance. The interframe background acquisition process could be improved, although the series of steps needed to modify the code was not implemented.

Overall the improved baseline reconstruction performance was measured to increase to about 7.5 models/second. Since basic video acquisition performance

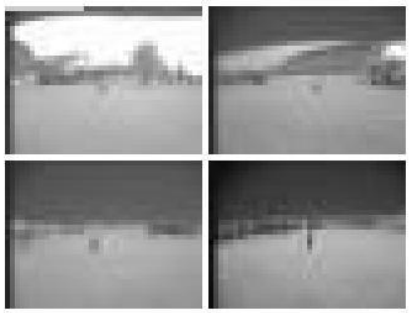


Figure 9: Low resolution views from each of the four modules.

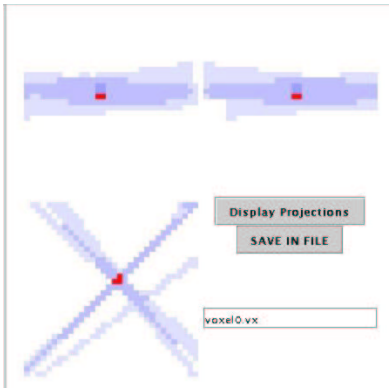


Figure 10: Projections along the x, y, and z axis from the 32x32x32 3D reconstruction.

is currently about 10 frames/sec we believe that some additional attention to module coordination would have allowed us to match this speed. Also, adding double buffering to acquisition software should increase both the video acquisition speed and the model reconstruction performance.

3.2 Tomographic Analysis Application

The analysis application was written to handle data streams from four sensor/processing modules. It was also enhanced to use prerecorded data sets so that the field trial data could be postprocessed after the acquisition.

Figure 9 shows the four frames used to reconstruct the volume. A dark set of pixels near the center of each image is the subject. The images are downsampled to 40x30 to generate the 32x32x32 volume model. Figure 10 shows the reconstruction of the scene. On the top of the panel are two orthogonal side views and beneath these are the top view. In each view, the progression in color from light blue to dark blue to red indicates the increasing overlap between module camera projections. The red area indicates the calculated position of the subject.

In addition to the tomographic implementation, we developed a display device for our system. Shown in the figure 11 is a commercially available PDA, the IPAQ pocket PC manufactured by Compaq Computers. This PDA is available with a wireless network link that provided us with the opportunity to link to our sensor modules. We wrote an application that displayed the video stream from any camera from any sensor module operating in our ground sensor network. With a modest enhancement this application would also be capable of displaying the tomographic reconstructions generated by the modules.

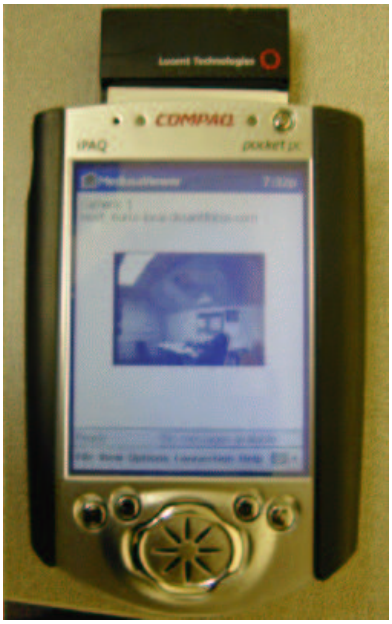


Figure 11: PDA displaying video acquired from sensor module.

3.3 ISL Visualization

We examined 3D reconstructions created from the field trial data, at visualization facilities within the Beckman Institute, with the assistance of Hank Kaczmarski and Camille Goudeseune. A series of 3D data sets were successfully transferred to Beckman and the data read into the volume rendering application. We viewed the volumes on a workstation and found that they virtually matched the projections generated by the tomographic analysis program.

The second field test data sets were inspected using the ALICE (Adaptive Laboratory for Immersive Collaborative Experiments) visualization environment. ALICE is a Beckman Institute resource at the Universisty of Illinois where the user enters a six-walled display cube and interactively controls the reconstruction display. We were able to view the tomographic reconstructions from a variety of angles, controlling the time sequential replay of the event. Figure 12 shows examples of a display frame from one of the visualizations.

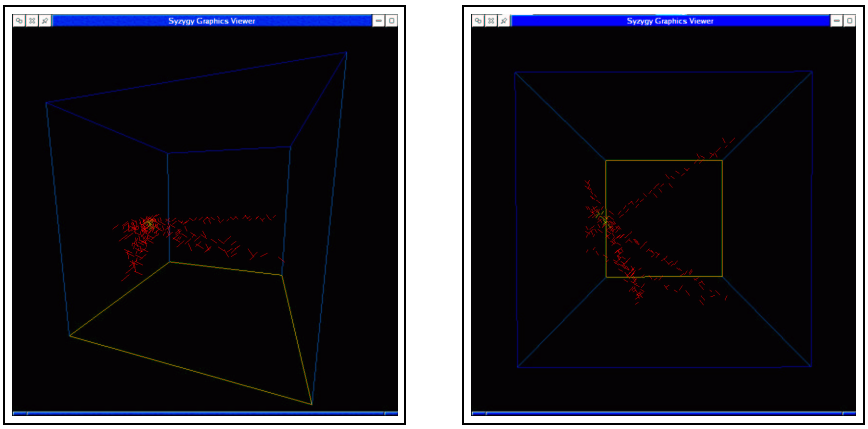


Figure 12: Two perspectives of reconstructed scene at University of Illinois' advanced rendering facility, ALICE. The yellow regions at the intersection indicate the location of the target.

Figure 13: Infrared images from two camera angles during field trial.

4 Field Tests

4.1 First Set of Field Tests

The first field trial of the unattended ground sensor array was performed in December, 2000 and is reviewed in the document "Report on System Field Test I"; attached in the appendix of this document. The test deployed two modules with attached infrared cameras and examined location and tracking of a test subject using triangulation.

A video frame from one test from each of the two modules is shown in figure 13. The calculated trajectory is shown in figure 14. A second live tracking test was also performed indoors using the visible spectrum CMOS cameras on two

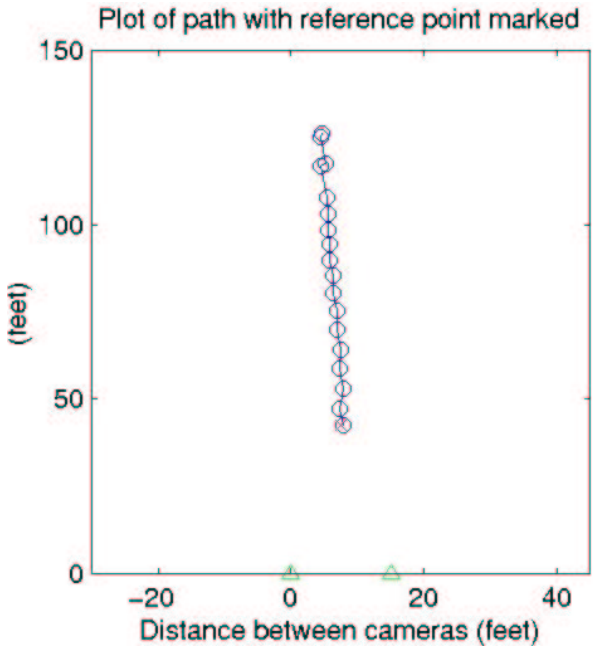


Figure 14: Range tracking during first field test.

modules. The comparison of the predicted location against the known location is shown in figure 15. Additional details can be found in the report.

4.2 Second Set of Field Tests

Data from two field trials were collected during August, 2001. The first field trial occurred on the afternoon of August 21 while the second occurred on the afternoon of August 23. Four sensor/processing modules were set up in the level area of a park directly behind the Business and Technology Center building at 701 Devonshire, Champaign, IL. The sensor modules were positioned on tripods at the four corners of a square measuring 50 feet on a side with camera number 1 aimed at a reference location in the center of the square. The location of each module relative to the others was measured. All cameras were positioned at approximately the same height. In addition, a series of test paths were marked and measured prior to the tests. In these tests, the monochrome CMOS cameras were used to acquire images (during the December, 2000 tests, a pair of IR cameras were used for image acquisition).

It became immediately apparent that the CMOS cameras would not perform adequately in direct sunlight. The intense sunlight and sky saturated most images and led to pixel bleeding. The problem was resolved for this trial by attaching temporary shades to the cameras and by raising the modules 3 feet

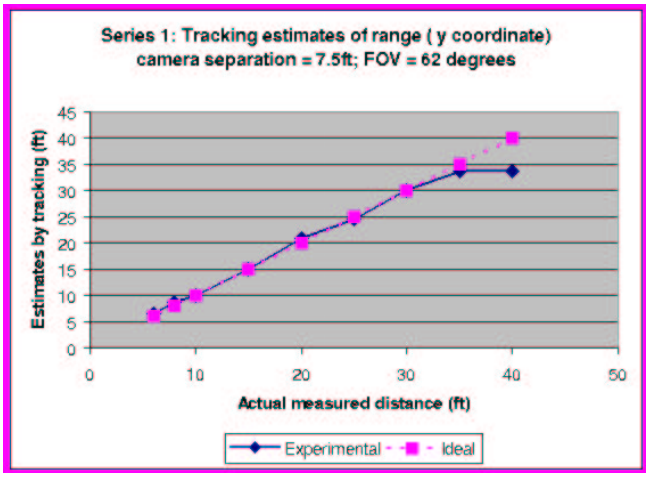


Figure 15: Live tracking via triangulation during first field trial.

higher and aiming them slightly downward toward the reference point. In this manner, direct sunlight and most of the view of the sky was eliminated from the image. A set of polarizing filters was attached to each camera for the second field trial to eliminate the brightness problem.

Several data sets were collected, lasting about 20 to 30 seconds each. In each test a human test subject walked at a constant pace from one reference point to a second and then back. Test data was collected for north/south, east/west, and two diagonal paths. In addition, a random path and a test with two subjects were also collected.

In the second data collection, on August 23, 2001, once again the modules were set up in the park in roughly the same layout as for the previous trial. Several of the tests performed in the first trial were repeated and a new set, with the subject tossing a large object up in the air as he walked, were acquired.

Several time-sequential tomographic reconstructions are displayed in figure 16. In this specific test, a human test subject started at a reference point at the north end of the test field, walked toward the south until they reached a second reference point and then turned around and returned. The distance between reference points was 40 feet and the test took slightly less than 20 seconds to complete.

Shown in figure 16 are projections of the 3D reconstructions generated every two seconds during one test. Each smaller figure is composed of a projection along the x axis looking north to south (top left corner), a projection along the y axis looking east to west (top right corner) and a projection along the z axis from above (bottom right corner). The areas are shaded in relationship to the intersection of individual camera models, i.e. red or dark indicates agreement between all three modules, dark blue is an intersection of two camera views, and

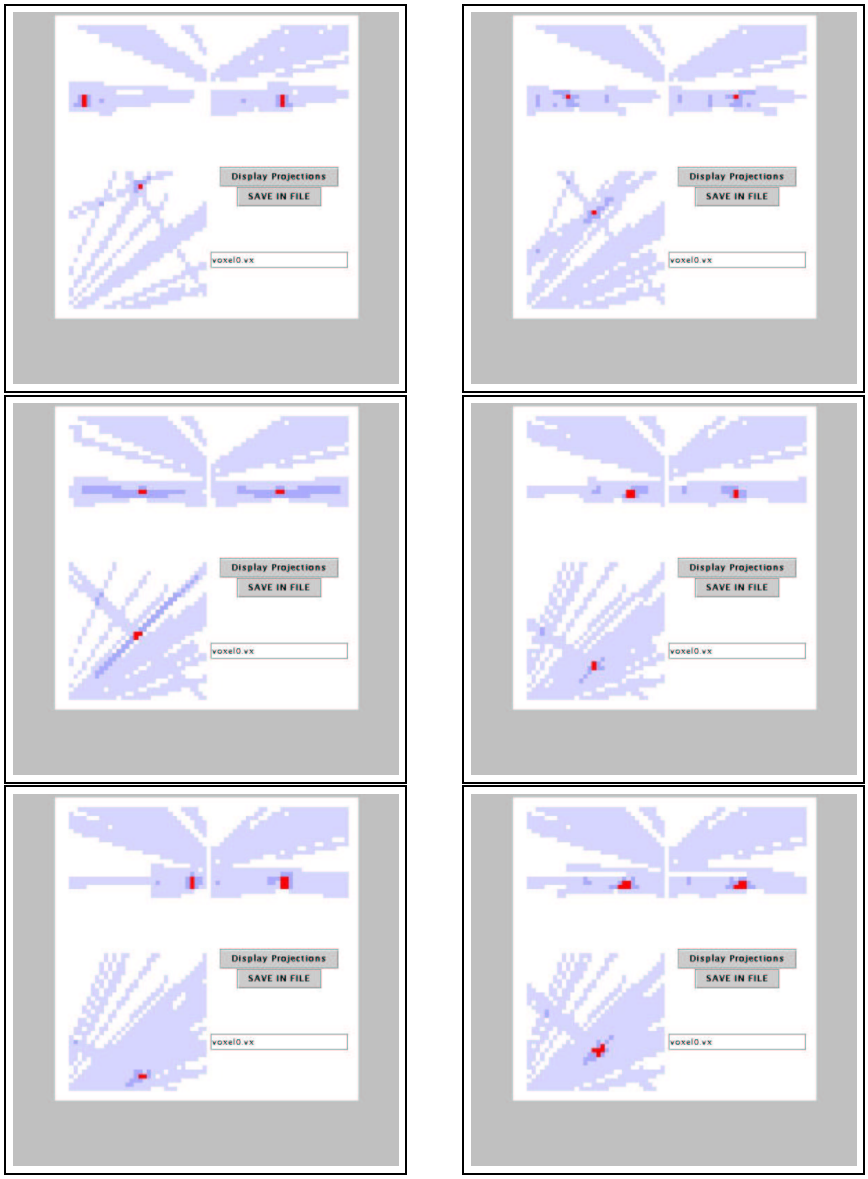


Figure 16: Time sequential 3D reconstruction of field test data showing x-, y-, and z-axis projections at 2 seconds intervals. Dark/red shaded areas are high probability zones for location of test subject.

light blue is the contribution from a single camera. It should be noted that there are extensive light blue areas in the top regions of the x and y projections. This is due to 'noise' created by clouds moving into the view of one of the cameras. Even though a large amount of this 'clutter' exists, the subject is successfully tracked up until just after this sequence. At this point, the sky clutter or other issues with camera orientation and distortion lead to problems in isolating the test subject.

A quick summary of the experimental results leads to the following observations:

- The accuracy of the locating the subject was, to first order, dependent on the size of the voxel or volume element. In these tests, the voxel was approximately one foot on a side. Higher resolution could be attained by using higher resolution images and subdividing the reconstruction model elements or by employing multiresolution techniques. The drawback of using larger models is that reconstruction slows appreciably unless the software is restructured. Multiresolution techniques limited to regions of interest show promise, but were not implemented in this system.

- Video rates are readily achievable. The two primary issues are processing the individual models and then communicating the models to a central display. So far, the bandwidth of communicating the models has been the primary bottleneck. However, we feel that by coordinating data exchange between individual modules so that only specific 'hits' are examined, the data load can be significantly reduced and video rates with models of $64 \times 64 \times 64$ and $128 \times 128 \times 128$ can be provided.

- The configuration of the cameras is important. In this test, the four modules were all placed close to the same planar surface. Placing one camera outside of this configuration would have added considerable discrimination in determinating the number of test subjects.

- Higher resolution models require considerable effort in locating and orienting the cameras and understanding their optics. Positioning the cameras quickly and precisely requires further study. We used a tape measure (accuracy of about 1"), however, this would begin to prove inadequate for higher resolution or larger scale trials. In addition, the current CMOS cameras formed distorted images at the larger angles. This could be compensated for during the geometry determination phase with little or minor additional computational overhead during model reconstruction.

5 Presentations and publications

Some of the work discussed in this final report has been published in the following conference proceedings:

R.L. Morrison, R.A. Stack, and D.J. Brady, "Insights into the development of two generations of networked sensor array and processor systems," in Integrated Computational Imaging Systems, OSA Technical Digest, (Optical Society of America, Washington DC, 2001), pp. 57-59.

A.M. Rittgers, R.L. Morrison, R.A. Stack, and D.J. Brady, "Tomographic processing on wireless ground sensor networks," in Unattended Ground Sensor Technologies and Applications III, Edward M. Carapezza, editor, Proceedings of SPIE Vol. 4393, pp 122-128 (2001).

R.L. Morrison, D.J. Brady, A.M. Rittgers, and R.A. Stack, "Wireless Integrated Sensing, Processing and Display Networks for Site Security," in Enabling Technologies for Law Enforcement and Security, S.K. Bramble, E.M. Carapezza, L.I. Rudin, Editors, Proceedings of SPIE Vol. 4232, pp 352-358 (2001).

This work was also the foundation for a thesis submitted for a Master of Science degree in Electrical Engineering at the University of Illinois - "Wireless Sensing and Processing Networks," by Andrew M. Rittgers (2001).

Finally, a "Report on System Field Test I" prepared by A. Rittgers, discussing the initial field test of the first generation sensor modules, is also included.

Insights from the development of two generations of networked sensor and processor systems

R.L. Morrison, R.A. Stack, and D.J. Brady
Distant Focus Corporation, 701 Devonshire MC-17, Champaign, IL 61820
phone: 217-366-8366, fax: 217-352-2446, morrison@distantfocus.com , http://www.distantfocus.com

Abstract: A wirelessly networked array of integrated sensor and processing modules has demonstrated video-rate tomographic reconstruction previously shown on a video enabled supercomputer cluster. We discuss issues regarding design and development of these first- and second-generation distributed sensing and processing platforms.

OCIS codes: (150.6910) Three-dimensional sensing, (100.6950) Tomographic image processing

Introduction

Our environment is filled with many fixtures using autonomous or semiautonomous sensing devices (e.g., motion sensor lighting, thermostatically controlled heating and cooling, smoke detector and CO alarms, sonically coupled automatic door openers, video surveillance, vehicle actuated traffic signals, and police radar.) As CMOS imaging chips and embedded processors continue to drop in price and wireless communications becomes ever more pervasive, the next level of smart sensors and distributed sensor arrays will be deployed to monitor and control our environment with ever increasing refinement.

As part of a DARPA funded project to examine the field deployment of a distributed array of ground sensors, we have designed and developed two generations of integrated sensor/processor module platforms [1,2]. The system objective was to incorporate sufficient sensing and analysis capability throughout a networked system to achieve tomographic volumetric modeling similar to what had been achieved using a supercomputer cluster in a related project [3]. In addition to the challenge of integrating sensor and processing components, we were able to explore issues of power consumption, power storage, non-volatile data storage, wireless network connectivity, and network security. Finally, designing the second-generation system with an eye toward commercialization also decreed specific requirements.

The tomographic analysis that was implemented was similar to many aspects of the basic cone-beam [3] or silhouette-style algorithm. Each module acquired a video stream from a sensor head composed of four CMOS cameras. The video stream(s) could be filtered and compressed, background subtracted and downsampled to lower resolution images. During tomographic analysis, the processed images were backprojected through a volume of interest. Finally, these individual volumes were collected from the distributed modules and combined to form a three-dimensional model of the local environment. This problem is interesting because of the large-scale processing and the potential high bandwidth intercommunications of the distributed system. This experiment also provides a test bed for examining the interplay of observers with this rich, immersive sensor field.

Platform Development

Of primary importance for the first generation system was the rapid development of the module. This enabled immediate transfer of tomographic algorithms originally developed for a studio deployed system [4]. Four video sensor head modules, an interferometric sensor head module, and a development environment module were built.

In order to speed prototype development on the first generation system, the computational elements were assembled from commercially available PC-104 standard components. A 233MHz Intel Pentium processor formed the core of the module processor board. Attaching enhancement boards incorporated additional features. For example, a video frame-capture board supporting four video channels connected with the sensor head containing four CMOS cameras. A PCMCIA adapter board supported 802.11b standard wireless network cards. A Compact Flash memory adapter provided access to large scale nonvolatile memory. One packaged module is shown on the left-hand side of Figure 1.

The ability to quickly and easily customize the operating system kernel, the availability of hardware drivers and open source code, plus the stable and solid performance of the Linux OS significantly streamlined code development. The Java development environment together with a web centric approach to feature development allowed us to rapidly develop user and display interfaces. Although the modules were typically operated with an AC adapter, the system's 10-Watt power demand drained the NiMH batteries within about an hour. We did find, however, that the Pentium design provided more than sufficient processing power for these experiments. Typically, the application required no more than 25-50% of the available processor instruction cycles. Several design rules were identified during the course of first generation module construction. We identified the following list of critical system issues:

- processing power - must be sufficient for image processing and potentially tomographic modeling
- power consumption - viable platforms should operate for many hours on battery power
- power storage - extended lifetime with state of the art NiMH and Lithium/Lithium-ion batteries
- sensor integration - use established USB, PCMCIA, and Compact Flash memory interfaces
- wireless digital network connectivity and security - link platform to users entering the sensor field
- software development platforms - Linux, Java, and other web enabled open source platforms
- nonvolatile data storage - flash or Compact Flash memory as opposed to hard disk drives
- system cost - objective of under $1000 per module cost

The goal of constructing a commercially viable surveillance system has had a great influence on the design of the second-generation sensor/processing module platform. Various embedded processors were evaluated for their processing capability, power consumption, and general system integration aspects.

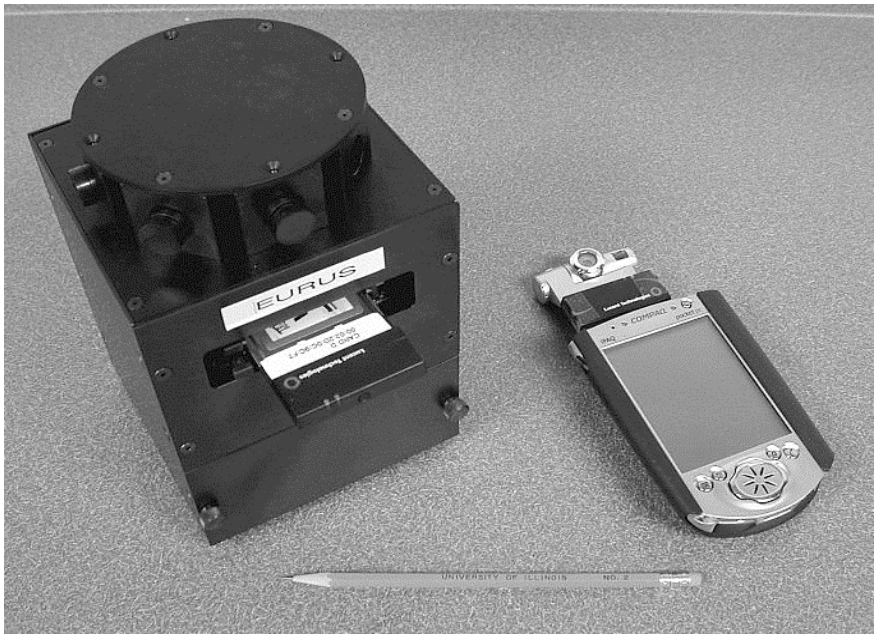


Figure 1 - First generation sensor module and second-generation prototype. On the left is the Pentium-based module with four CMOS video cameras in the sensor head. On the right is a mockup of a StrongARM system with wireless network card and single CMOS camera (based on Compaq iPAQ PDA).

The emerging retail market for low-cost, low-power personal digital assistants has produced many advances that can be incorporated into the processor core of the integrated sensor and processor module. One of the initial candidate systems for the second-generation platform was a retail personal digital assistant (PDA). The Compaq iPAQ model 3600 series handheld PC has several desirable features. The iPAQ is based on the StrongARM processor, which operates at full performance on less than a Watt of power and can rest in standby mode at sub-milliwatt levels. By adding an expansion pack for PCMCIA cards, the module can be outfitted for short-range digital

wireless networks. Unfortunately, the iPAQ's USB connectivity is limited to slave mode making it incapable of controlling standard USB video and audio sensors. Also, the LCD display is a nonessential element contributing to power consumption and additional system cost. Still, the basic unit cost of between $500 and $1000 and the small-scale package illustrate the potential for our customized system.

Our second-generation module shares many aspects of the iPAQ PDA, but was designed independently using an extended feature StrongARM system development kit. It uses the Intel StrongARM processor (SA-1110) operating at 206 MHz and incorporates the SA-1111 for host USB control. The StrongARM platform is available with WindowsCE and Linux OS support. Shown on the right side of figure 1 is a mockup of a StrongARM prototype based on the iPAQ form factor and a Compact Flash module integrated CMOS camera. The reduced size and significantly extended lifetime suggest many opportunities for this platform. In addition, press releases from Intel indicate that 600MHz operation at less than half watt power consumption will soon be achieved with this processor microarchitecture. The StrongARM system chip set includes the SA-1111 for controlling communications between USB devices that forms the basis of the sensor head. USB has rapidly established itself as an industry recognized standard for exchanging digital data between audio, video, and other moderate bandwidth devices. This development platform is tightly integrated to options for Compact Flash (nonvolatile storage), micro drives (miniature hard disks), PCMCIA interface (for wireless/wired networking and modem connectivity), advanced battery management circuitry, plus other hardware enhancements typically targeted to the mobile laptop PC market.

One potential drawback of using a StrongARM processor based platform is the lack of a well integrated hardware floating point coprocessor. Floating point calculations are typically carried out via software emulation, sometimes with significantly reduced performance. Care must be taken with time critical software to operate primarily in integer mode.

Our goal with the second generation platform is to construct a large system (more than 12 modules) in order to explore practical aspects of deploying tomographic modeling outside of a laboratory environment. In addition, we plan to offer this platform as a commercial surveillance system.

Summary

Advances in power critical processing driven by the mobile laptop and PDA market provide remarkable opportunities for developing compact integrated sensing and processing platforms for autonomous remote monitoring. We have developed considerable expertise while designing two platforms that demonstrate sophisticated tomographic analysis. Hardware, software, and networking issues are being rapidly addressed, thereby setting the stage for deployment of highly evolved sensing and processing platforms.

Acknowledgements

This work was supported through DARPA's Tactical Sensor Program via Army Research Contract DAAD19-00-C-0099. The technical point of contact for this DARPA program is Dr. Edward Carapezza.

References

1. R.L. Morrison, D.J. Brady, A. Rittgers, and R.A. Stack, “Wireless integrated sensing, processing, and display networks for site security,” SPIE conference 4232, Technologies for Law Enforcement and Security (2000).

2. A. Rittgers, R.L. Morrison, R.A. Stack, D.J. Brady, "Tomographic processing on wireless ground sensor networks," to be published in SPIE conf. 4393, Unattended ground sensor technologies and Applications III, (2001).

3. L.A. Feldkamp, L.C. Davis, and J.W. Kress, "Practical cone-beam algorithm," J. Opt. Soc. Am. A, pp. 612-620, 1984.

4. D.J. Brady, S. Feller, E. Cull, D. Kammeyer, L. Fernandez, R. Stack and R. Brady, "Information flow in streaming 3D video," to be published in SPIE Critical Review of Three-dimensional Video and Display, Photonics East 2000.

Tomographic Processing on Wireless Ground Sensor Networks

Andrew M. Rittgers

Department of Electrical and Computer Engineering, Beckman Institute, University of Illinois at Urbana-Champaign, Urbana, IL 61801

Rick L. Morrison and Ronald A. Stack

Distant Focus Corporation, Champaign, IL 61820

David J. Brady

Department of Electrical and Computer Engineering, Fitzpatrick Center for Photonics and Communications Systems, Duke University, Durham, NC 27708

ABSTRACT

New opportunities for battlefield surveillance and modeling are unfolding with the advent of smart sensors linked via digital wireless networks. One exciting prospect is the use of tomographic techniques in order to create real-time three-dimensional modeling and analysis of the environment that is immediately accessible to battlefield forces. We have developed a small-scale ground sensor network for this application. We discuss initial deployment of this network as a tracking system.

Keywords: sensor networks, interferometric sensors, ground sensors, tomographic imaging, wireless networks

1. INTRODUCTION

A set of smart sensors called the Medusa Network was developed by the Photonic Systems group at the University of Illinois for the purpose of developing a platform for tomographic analysis on distributed wireless ground sensor networks. The catalyst for creating the Medusa Network came largely from work done on the Argus Distributed Sensing and Processing Environment at the University of Illinois [1]. Argus is a Beowulf-class parallel computer designed as a test-bed for three-dimensional (3D) imaging using distributed processing. The environment consists of a circular sensor space 14 feet in diameter surrounded by 64 video cameras. Pairs of these cameras are connected to each of the 32 dual-processor Linux machines that make up the Beowulf cluster. The cluster is capable of generating a 3D voxel array, 128 elements on a side, at a rate of two "frames" per second, however, modest improvements in software and system hardware should speed the reconstruction rate up to eight frames per second.

By "tomographic analysis" we mean the reconstruction of multidimensional scenes from projection data. Computed tomography (CT) is used most often in x-ray reconstruction of translucent 3D objects. However, as discussed in Reference 2, algorithms can be applied without modification to 3D reconstruction of opaque visible objects. Digital scene analysis and surface abstraction may be added to CT algorithms to analyze opaque objects. CT algorithms are a related subset of computer vision scene analysis tools. In computer vision one may choose to logically analyze scenes from single perspectives and then to logically fuse scene interpretations over temporally and spatially distributed frames or one may choose to physically integrate frame models to form a multidimensional target model prior to logical analysis. The target model might consist of 3D models of objects distributed across a plane. This paper considers abstraction of object position across the plane from sensor array data. While back projection for target positions is a very weak form of tomography, we refer to this physical space reconstruction as tomography to contrast it with logical frame analysis.

Tomographic analysis allows targets to be analyzed in their native 3D or 4D spatio-spectral spaces and removes many of the ambiguities of conventional two-dimensional (2D) analysis. As the angular range of the captured target data is increased, tomographic analysis becomes increasingly more effective. The angular range can be increased by tracking relative motion between the sensor and the target and by cooperative target analysis across a sensor network. We consider both approaches and focus in particular on distributed tomographic analysis, image analysis and target abstraction algorithms. To date we

have developed and tested the object detection and tracking capabilities of the network. In addition to discussing the motivation behind distributed tomographic sensor networks, this paper also describes the construction of our first functioning sensor network and the field tests performed on this network.

2. TOMOGRAPHIC ALGORITHMS

One of the more difficult tasks in building a networked array of ground sensors is that of gathering information from the distribution of sensors and then integrating that information for the purpose of tracking and target identification. Related efforts have studied network algorithms and data fusion [3,4]. One of the issues found in these studies centers on network granularity. Granularity refers to the sensing and processing capabilities of each node within the network. In a system with fine granularity, sensors communicate all the information they detect to a central processor. In a coarse grain system, target classification is implemented at the sensor level and sensors communicate classification results. Thus the sensor array could be used to combine raw sensor signals into a global model before attempting target analysis (fine granularity data fusion) or the array could be used to combine locally-produced target analyses into a global analysis (coarse granularity data fusion). The fine grain approach requires substantial data transfer between sensor nodes. The coarse grain approach requires substantial processing power and memory at the sensor nodes.

A central design issue when implementing target analysis on a sensor network is determining what level of granularity one should assign to the sensor and processor resources. Tomographic algorithms provide a natural basis for completing this task. To accurately choose the appropriate level of granularity, we must determine how to optimally distribute the computation among the sensors and a central processor. In practice, one could implement a hierarchy of algorithms that form tomographic models on disparate data types (source intensity and target probability densities are example data types). In such a hierarchy, low-level algorithms would form local models based on measured signal intensities. Higher level algorithms would form probability models based on local processors' target identification. Now the design question ultimately reduces to "How should processing and communication be balanced at each level of the processing hierarchy?" The answer to this question depends on several factors. For example, sensor density is critical. On very sparse sensor arrays, the information received by each sensor is likely to be independent of the other sensors. In this scenario, a coarse grain approach is suitable. As one improves array resolution by increasing the sensor density, however, common processing of array data is increasingly attractive. We claim that fine to moderate granularity approaches, which emphasize low-level communication, are more efficient on dense arrays because they can be integrated into array hardware, thus reducing the need for general-purpose central processing.

A second critical design issue of sensor arrays is network structure. Traditionally, a central processor gathers information from sensor nodes and combines the information in an optimal or efficient manner. While it is clear that a network with a star topology (all sensors connected to a central processor) will be able to extract a maximum amount of information from the collected data, this approach has a number of serious drawbacks. For example, the aggregate communication necessary between the sensors and the central processor is a bottleneck due to the large amount of necessary bandwidth and interference in a wireless scenario. In addition to these required communications resources, the central processor must have sufficient computational resources to assimilate all of the collected data. An alternative to such a centralized network is a distributed network, in which integrated sensors and processors communicate as peers. Although the central processor approach is easier to program and conceptualize, it is also less robust against processor failure and requires significantly more power and processing capacity in a single location. Under the distributed approach, local processing may be included in the sensor design and the network topology can be designed to enable tomography and classification through iterative belief propagation of simple, locally computed, information. Eliminating the need for global communication is an appealing potential feature of distributed sensor networks; e.g., in wireless ad-hoc networks, sensors could use power allocation to adjust their transmission power until links to a relatively small number of neighboring sensors are established. Such a scenario would mitigate both power consumption and multiple-access interference.

Considering the points made in the preceding paragraph, choosing the right topology for a particular application may be just as important as how one implements it. One common tomographic algorithm that we explore uses convolution and backprojection [5]. The convolution step weights the output of each sensor based on its orientation to the spatial points of interest. The backprojection step sums the values weighted to produce a source density at each point. This exact approach can be used to combine target probability densities. In addition to the standard convolution and backprojection method, we choose to explore silhouette reconstruction as a less computationally expensive method of tomographic reconstruction. Silhouette reconstruction is a binary method where a sensor contributes a yes or no response as to whether an object is

present at a specific location in the scene. This method can be used as a quick means to determine regions of interest before more costly reconstruction methods are used. Practical implementation of these tomographic algorithms within our sensor module array will use distributed processing and distributed control to achieve model reconstruction. Inter-module data communication will be dramatically reduced through efficient analysis and by limiting exchanges to hypothesis verification and/or resolution enhancement. The initial scenario employs up to four sensor modules, each one equipped with four cameras, to monitor a scene. By limiting the reconstruction volume to a coarse resolution, the system can achieve a throughput of a few models per second. Also, by embedding the control throughout the distributed network rather than within a centralized control station, wireless laptops and personal digital assistants (PDAs) can easily connect with the secure network and immediately request information and displays of the reconstructed environment. Continued enhancements will eventually lead to event triggers where the sensor modules will identify specific activity and request user intervention.

3. SENSOR MODULES

A network consisting of four prototype sensor modules was constructed at the University of Illinois and deployed on a trial basis for evaluating sensor array operation. For this first network, each module, or node, was constructed using off-the-shelf commercially available PC-104 components. Future networks will employ a custom module design and should be expected to conform to higher standards of compactness and power conservation. The PC-104 platform provided a computing standard that conformed to our size requirements and could be rapidly developed with little modifications to software and accessory hardware. A photo of the module without its case is shown in Figure 1.

The core of the module is a PC-104+ 266MHz Pentium processor board with 64M of onboard RAM. The board has power saving capabilities similar to those found in laptops, such as throttling the clock speed of the CPU during idle periods. A PC-104+ frame capture card with four multiplexed input channels was used to acquire images from four CMOS cameras. Stitching together the images from the four cameras, the sensor modules have the added feature of being able to generate 180-degree panoramic views. Interfaces were also designed to allow inputs from the infrared cameras used in the field test. A PCMCIA socket board integrated IEEE 802.11b wireless ethernet cards into the system. Finally, a 96MB CompactFlash module was used for data and system storage. Each module was packaged in a custom designed anodized aluminum housing, which includes the 180-degree array of the four CMOS cameras and the power supply. Using eight rechargeable NiMH “C” sized batteries, each module can operate for more than an hour. By optimizing power management functions, that duration could be increased. Each module can also operate on an AC power supply for long-term development and testing purposes. As mentioned above, the prototype device was not designed for maximum power savings, and future custom designs should improve these values up to ten-fold.

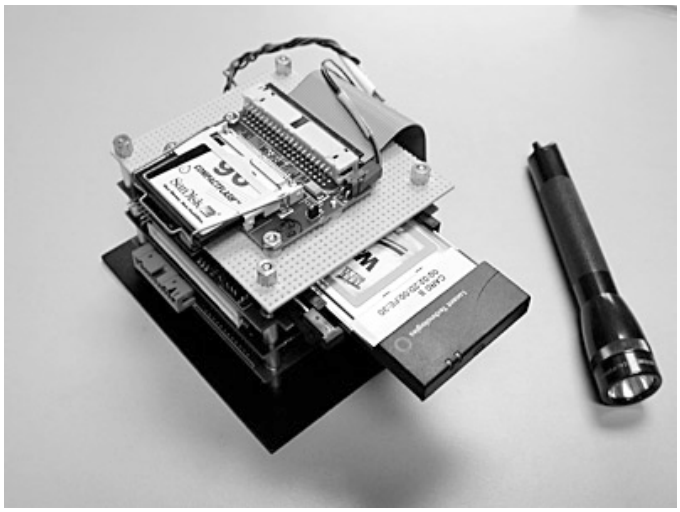


Figure 1. An exposed module is shown next to a 3-inch flashlight.

Each module was driven by a reduced version of the Linux operating system. Based upon common distributions publicly available at the time, we were able to reduce the operating system to a size less than 20MB, running only those functions essential to the operation and stability of the network. Several server-side software applications were custom developed by the group and installed on the modules. Included in this list is an image processing and module control application that provides connectivity through software sockets. Example processes include image acquisition and transmission, background subtraction, panoramic view creation, and image compression. The modules can serve multiple requests by forking each process to a new thread for each connection. On the client side, a graphical user interface (GUI) was written in the Java programming language. Basic Java applications benefit from being easily integrated with web browser applications and relative platform independence, making Java an important factor for integration within a heterogeneous sensor network. While current implementations of the interface operate on desktop or laptop computer environments, we seek to expand the interface to devices such as personal data assistants and other display devices.

4. EXPERIMENTAL SETUP

A primary goal of the project is to demonstrate, using prototype sensor modules, that tomographic data fusion is feasible using existing technology. We defined the scope of the first system field test as using tomographic analysis between two sensor modules to estimate the range and velocity of the specified target. At the time of the test, the network provided both real-time analysis and systematic collection of data for post processing. The latter was used to assist with software enhancements and verify the accuracy of results. Future work on the network will involve automating more of this data collection and processing. The test site had limited space available thus only ranges from 30 to 165 feet were used in the testing. To maximize the accuracy of tracking, we used the IR cameras connected to the sensor modules and a human subject in this test as illustrated in Figure 2. Due to a relatively large contrast between the subject and surrounding environment, the subject easily stands out and eases the process of determining the position of the object in the frame. Due to the limitations for determining precise distances and accurate sensor module orientation from physical measurements, we used a method of digital alignment, whereby we record a "reference" frame that captured the subject at a known and accurately measured location. From the data obtained from this reference frame, we were able to make range estimates of the object as it moved throughout the object space.

Figure 2. Sample IR image from test.

The first test involved two modules separated by about 16 feet. Data was collected for this distance as the subject moved in the sensor space on a preset course. The test was repeated with the separation between the cameras increased to about 82 feet. The movement of the subject was restricted to a single path perpendicular to the base line of the sensor modules, intersecting at a median distance between the sensor modules. This restriction insured that the test was repeatable, allowed us to make accurate measurements of the path, and limited the overall amount of error present in the setup of the test.

5. PROCESSING AND ANALYSIS

We were able to demonstrate object tracking on the 2D plane described by the location of the cameras and the location of the subject. This was accomplished by a triangulation method similar to a restricted tomographic analysis. This method involved the identification of the object, the location of its centroid, and calculations to estimate its position in the field. Calculations were first conducted on the reference images, and subsequently on the data images. Data from subsequent images was then used to calculate an average velocity of the object.

The method of calculating the object location from the camera images is illustrated in Figure 3 and 4. This is accomplished by defining the line equations of the rays that travel between the object and the sensor module. The parameters ϕ and ϕ' are calculated with the following equations,

$$\phi = \tan^{-1}\frac{x}{y}, \phi' = \tan^{-1}\frac{X-x}{y}$$

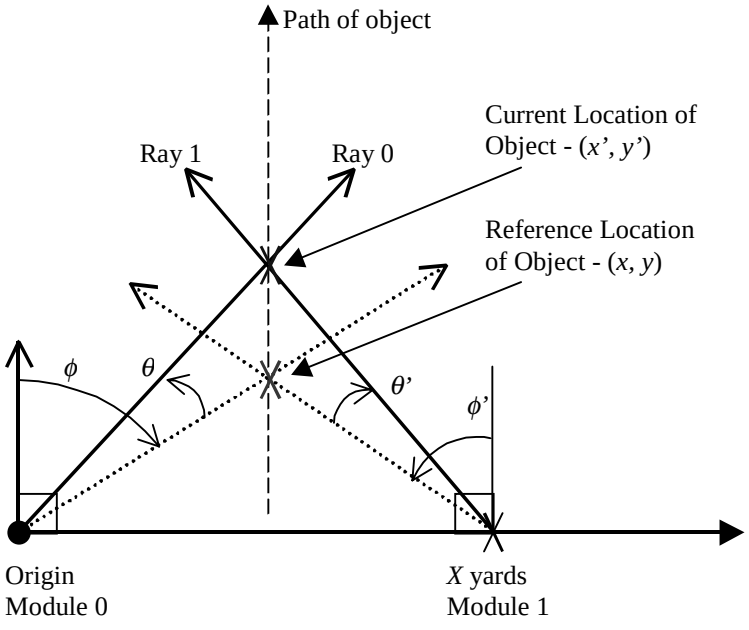


Figure 3. Experimental setup as seen from above

where (x, y) are the coordinates of the reference location, and X is the separation between the cameras. Each subsequent data image yields new values of θ and θ', thus revealing the location of the object relative to the cameras and reference frame. The values for these parameters are calculated with the following equations,

$$\theta = \frac{Px_0}{Ntotx} \times FOV(^\circ),$$

$$\theta' = \frac{Px_1}{Ntotx} \times FOV(^\circ)$$

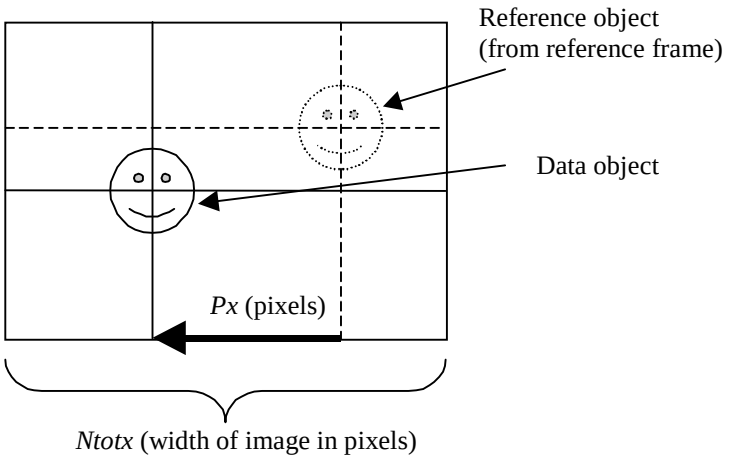


Figure 4. Measurements made from image frame

where Px represents the difference between the centroids of the current object location and the reference object location, ($Px = C_{obj} - C_{ref}$), $Ntotx$ is the width of the image in pixels, and FOV is the field-of-view of the camera in degrees. The slope of ray 0 (m_0) yields the ratio of the coordinates we are interested in (x', y'), which is dependent upon θ and ϕ. A similar calculation can be made on ray 1 (m_1) using θ' and ϕ'. Both slopes are calculated with the following equations,

$$m_0 = \frac{y'}{x'} = \frac{1}{\tan(\phi + \theta)}, m_1 = \frac{y'}{x' - X} = \frac{1}{\tan(\phi' + \theta')}$$

where values for ϕ and θ were calculated above, and X is still the separation between the cameras. Since the cameras are known to be on the lines describing rays 0 and 1, we use their locations to determine the y-intercepts of these lines (b_0 and b_1) according to the following equations,

$$b_0 = y_{cam0} - m_0 x_{cam0} = 0 \qquad b_1 = y_{cam1} - m_1 x_{cam1} = -m_1 X$$

where we define camera 0 to be at the origin, thus b_0 is zero, and the y values for both cameras are zero. We complete the range calculation by solving these simultaneous equations,

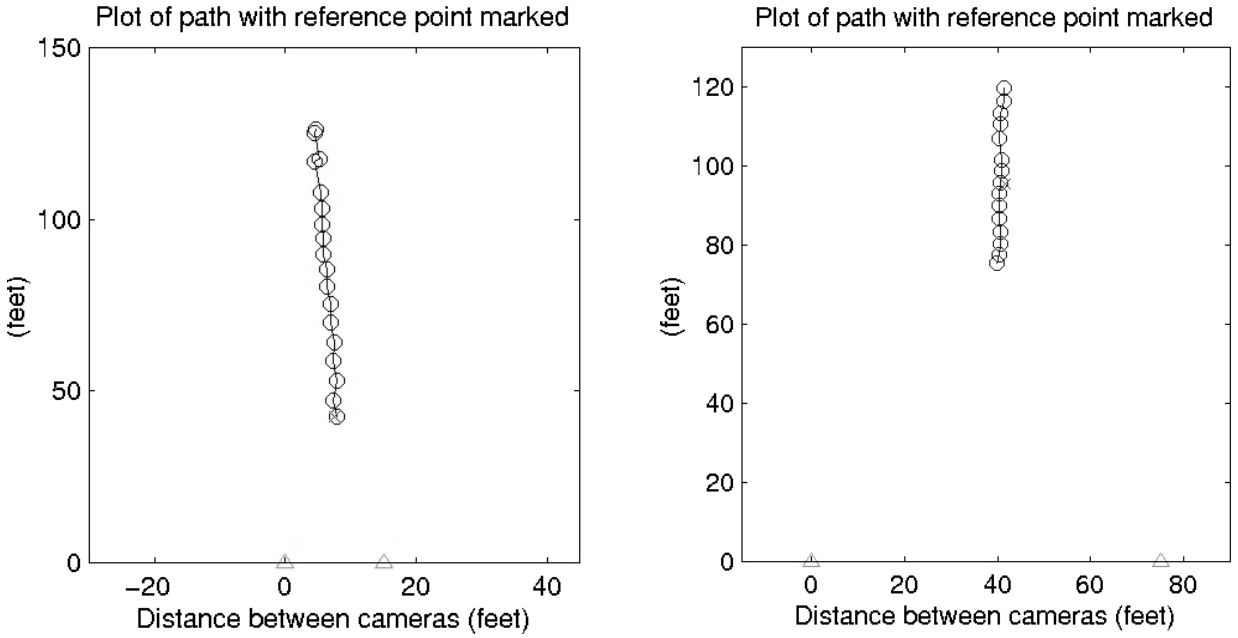


Figure 5. This figure shows plots of the output from the tracking algorithm on tests with camera separations of about 5m(15ft) and 25m(75ft). Missing circles are from frames that were skipped due to inadequate data.

$$\begin{aligned} y' &= m_0 x' + b_0 \\ y' &= m_1 x' + b_1 \end{aligned} \quad \Rightarrow \quad \begin{bmatrix} y' \\ x' \end{bmatrix} = \begin{bmatrix} 1 & -m_0 \\ 1 & -m_1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \end{bmatrix}$$

to get (*x'*, *y'*), the location of the object.

Results of the analysis, shown in Figure 5, are consistent with expectations described in the setup. The plots show the results of the two tests, one at a separation of 15ft, and the other at a separation of 75ft, as the subject walks almost directly away from the cameras at a distance halfway between them. Circles represent the path of the subject, triangles represent the cameras, and the small "*x*" represents the location where the reference image was captured. The circles appear at regular intervals, which is consistent with the subject walking at a regular pace. The average velocity of the subject was calculated to be 2.71 ft/s and 2.25 ft/s for each of the tests. These values are within the range of average walking speeds.

The third system test was performed using some software enhancements that provided real-time feedback on the tracking status. With a simple input of data collected from a reference measurement, we were able to track a bright object (for ease of object identification) with relatively accurate results. The test was conducted on a short-range basis to accommodate the indoor facility. Again, performing a two-dimensional object tracking, we set up the test with speed and simplicity in mind. The software acquired data by binning columns of pixels together. These bins were compared against one another in a winner-take-all fashion. The winning bin was the one that had the brightest overall value, and it was assumed that the object lay in this column. These coordinates were then entered into calculations similar to the triangulation method described above and range estimates were calculated. The results of this test are shown in Figure 6. It can be seen from this graph that the estimates conform to the ideal curve.

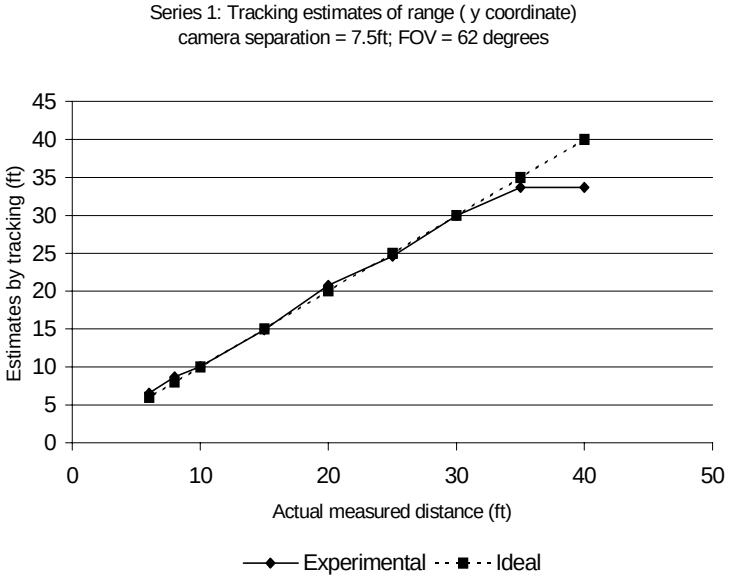


Figure 6. Plot of results from automated tracking algorithm.

6. CONCLUSION

The success of the Medusa Network largely depends on the ability to find the balance between a centralized and a granular approach to distributed tomographic processing. It is the goal of this research project to determine this balance point for object tracking using our ground sensor network. Tomographic analysis has many advantages over conventional imaging for target tracking. By reconstructing targets in their native 3D environments, tomographic analysis yields information not available to conventional 2D systems. With this additional information, ambiguities normally present in a conventional system can be resolved. As such, we will continue to develop more sophisticated tomographic algorithms to explore the

benefits of 3D and 4D analysis over conventional 2D tracking analysis. In addition to improvements in algorithm design, our sensor network will be enhanced as small electronic devices continue to see incremental improvements in computational speed, power consumption, and component cost. Future experiments involve tomographic methods of tracking multiple objects in a 3D volume. At first, this would be at very coarse resolution; however, as algorithms and communication protocols improve tracking resolution will become more detailed with greater accuracy.

7. ACKNOWLEDGEMENT

This work was supported through DARPA's Tactical Sensor Program via Army Research Office Contract DAAD19-00-C-0099. The technical point of contact for this DARPA program is Dr. Edward Carapezza.

8. REFERENCES

1. D. J. Brady, S. Feller, E. Cull, D. Kammeyer, L. Fernandez, R. Stack and R. Brady, "Information flow in streaming 3D video," SPIE Critical Review of Three-dimensional Video and Display, Photonics East 2000.
2. D. L. Marks, R. A. Stack, D. J. Brady, D. Munson, and R. B. Brady, "Visible Cone-beam Tomography with a Lensless Interferometric Camera," Science, **284**, pp. 2164-2166, 1999.
3. M. G. Corr and C. Okino. "A Study of Distributed Smart Sensor Networks," Dartmouth College, Thayer School of Engineering Technical Report Preprint, March 2000.
4. D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next Century Challenges: Scalable Coordination in Sensor Networks." In Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking, pp. 263-270. Seattle, Washington, USA, ACM. August 1999.
5. L. A. Feldkamp, L. C. Davis, and J. W. Kress, "Practical cone-beam algorithm," J. Opt. Soc. Am. A, pp. 612-620, 1984.

Wireless Integrated Sensing, Processing and Display Networks for Site Security

Rick Morrison, David J. Brady, Andrew Rittgers, and Ronald Stack

Department of Electrical and Computer Engineering and Beckman Institute for Advanced Science and Technology, University of Illinois at Urbana-Champaign, Urbana, IL 61801

ABSTRACT

We consider data management on *ad hoc* networks of sensing and processing nodes. We describe the construction of simple nodes from off the shelf components (PC 104 single board computers with flash memory, video capture cards and 802.11b wireless interfaces). We describe a Java interface to controlling these nodes and accessing images and image processing algorithms. We demonstrate target tracking across nodes and the potential for heterogeneous sensor types.

Keywords: Sensor networks, interferometric sensors, ground sensors

1. SENSOR SPACE SECURITY

We consider security in ubiquitous sensor environments. For example, the environment could be a building on which a dense network of visible and infrared cameras, laser scanners, magnetic, acoustic, smoke, toxin and temperature sensors have been deployed. The logical sensor network is mapped onto the physical environment. This network must satisfy the following constraints:

- Data security. Sensor data should be accessible only to authorized personnel but must be available quickly, easily and effectively to the security team.
- Data availability. Sensor data must be available in the site rather than just at a control room. Security personnel working in the site should have immediate and effective access to sensor data.
- Data specificity. The sensor network must respond to specific queries and trigger on specific events requested by the security team. Team members must not be forced to manually filter sensor data.

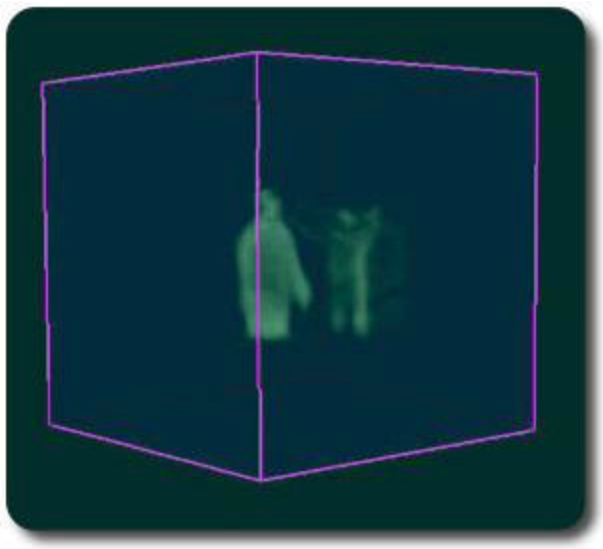


Figure 1. Projection of a 3D reconstruction from Argus.

These constraints can only be satisfied by embedding digital processors on the sensor network.

This paper considers wireless sensing and processing modules under development at the University of Illinois. These modules use a compact version of the Linux operating system to achieve flexible embedded processor/digital network devices with general purpose sensor ports, high power efficiency and collective programming. We describe cost, network density, secure interfacing, embedded processing, network robustness, power and deployment issues for dense installations of these modules and we describe results from an experimental demonstration of a sensor network.

Related efforts have previously considered data fusion on networks ranging from robots [1] to dust [2]. Several studies have previously considered network algorithms and data management [3,4, 5]. Key issues in this previous work center on network granularity. Granularity refers to the capacity of

sensing, processing and communications components at each node. This paper does not present a comprehensive review of the large literature of previous work in distributed sensors, sensor networks and sensor data fusion. Rather, we focus narrowly on the issue of simple data management on relatively granular networks based on previous work on 3D video recording studios.

The impetus for considering dense arrays of wireless sensors arises partly from previous results with the Illinois Argus sensor space [6]. Argus is a Beowulf-class distributed computer consisting of 32 dual-processor Linux machines interfaced in parallel to 64 digital video cameras. The Argus network sensor network is hardwired with 100Mbs ethernet switching between nodes. The camera array captures a 4.3 meter diameter circular sensor space. Argus is currently capable of generating approximately 2 frames/second of a 128x128x128 3D reconstruction of the scene in the sensor space. We expect incremental improvements in software to push the reconstruction rate to 8 frames/second on existing hardware. As an example, an Argus 3D reconstruction of a pair of martial artists facing off is shown in Figure 1.

One of the interesting lessons from Argus is that the centralized notion of sensor data fusion by building a complete environmental model is not always appropriate. Figure 1 was generated by projecting the Argus-generated data set on advanced SGI hardware. In many cases it is extremely inefficient to backproject the full set of sensor data to build a model and then to forward project the model for scene display. This inefficiency is particularly apparent in multi-user environments. In some environments, as when a number of far remote users want to observe a scene, it is efficient to build a full 3D model and then transmit the model for redistribution. In other cases, as in a number of near users observing a scene in which they may themselves be immersed, it is more efficient to map projections from the sensor field onto the user visual fields without forming a centralized model. This second situation is most applicable to the site security application.

Our focus here on situations in which the sensor space and the observer space are identical. The sensor space may be a building, campus or installation. A variety of CCTV and wireless sensor resources may exist in the facility. Observer-deployed, observer-carried and robot deployed devices may augment these resources. Our challenge is to effectively map data from these sensors onto observer demands. Users may make a variety of requests on the network, such as what is around the next corner, what is upstairs, where is the person in a red jacket, or even what was the person in the red jacket doing 5 minutes ago. We have successfully implemented interactive caching and space-time analysis on the hardwired Argus array. In this paper we consider transferring these techniques to wireless networks. The second section of the paper describes the hardware and software at the core of our sensing and processing nodes. The third section describes implementations of specific security, data fusion and scene analysis on the network. In the final section we describe critical challenges in extending these networks and suggests problems for further analysis.

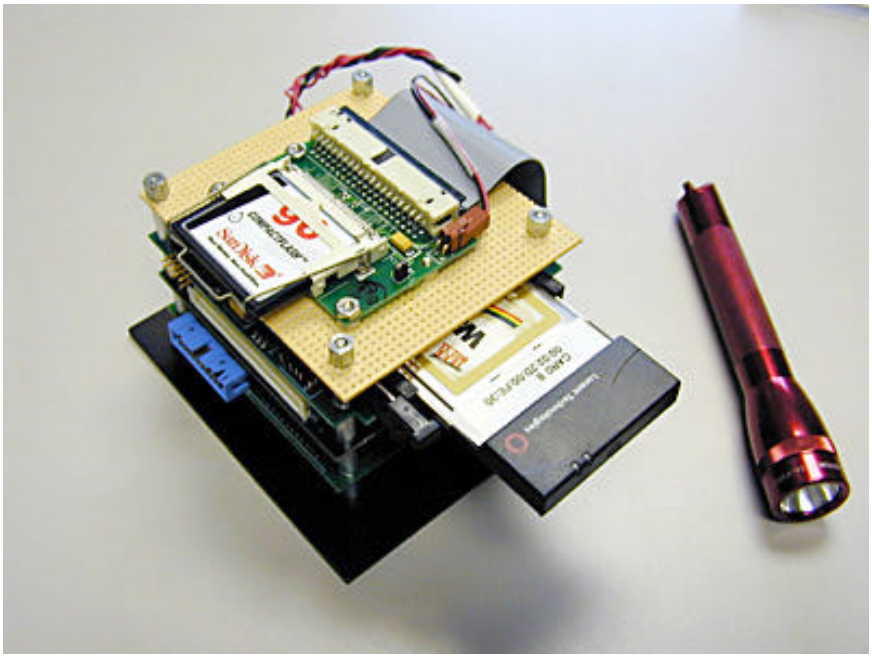


Figure 2. Sensor-processor node next to a 3 inch flashlight. The wireless card sticks out the front. The flash memory is on top.

2. SENSING AND PROCESSING NODES

The design and assembly of the sensor and processing nodes are impacted by many issues such as cost, power consumption, processing performance, network configuration and channel bandwidth. The goal of this phase of the project was to create a test bed for examining basic issues of sensor processing and data presentation. One can assume that a commercial design will

be more compact and power efficient. The sensor and processing modules described in this paper were assembled from standard commercially available components in order to rapidly develop the wireless sensor system. The PC-104 computer board standard was selected because the electronic boards have a small form factor, there are several special function boards, and multiple vendors sell components. A typical packaged sensor module is shown in figure 2. The closed sensor package is shown in figure 3.

At the core of the system is the microprocessor card that controls the various peripherals and runs the image acquisition and processing applications. A 266MHz Pentium processor board was selected because it provided the highest performance of currently available boards. In addition, this specific board supports a performance throttle feature that provides a means of idling processor operation and thus reduces power consumption during periods of low activity.

Generally, most desktop/laptop processing system use a hard disk for OS, application, and data storage. Hard disks consume considerable power, are subject to mechanical and environmental factors, and are far larger in capacity than needed by many of our applications. Instead, we selected to use Compact Flash memory cards for storage. These cards, typically used for digital camera image storage, are widely available in sizes up to several hundred megabytes and provide a low power, nonvolatile storage unit. An additional feature of Compact Flash usage is that, by interchanging programmed cards, the functionality of the module can be quickly changed.

A system video capture board was used to acquire images. Four video sources were connected to the acquisition board and an onboard multiplexer allowed rapid selection between input channels. Several of the sensor modules were configured with four monochrome CMOS cameras, mounted in a half circle, and attached to the video inputs. This configuration provided the capability of creating 180-degree panoramic images. Sensor modules in general could also be outfitted with infrared cameras, specialized interferometric sensors, and other thermal, audio, biochemical sensors.

Each module was designed with a PCMCIA adapter that provided the means of integrating an 802.11b standard wireless network card. With these cards, a 10 Mbps encrypted data stream was available for transmitting control, data, and images between sensor modules and display devices.

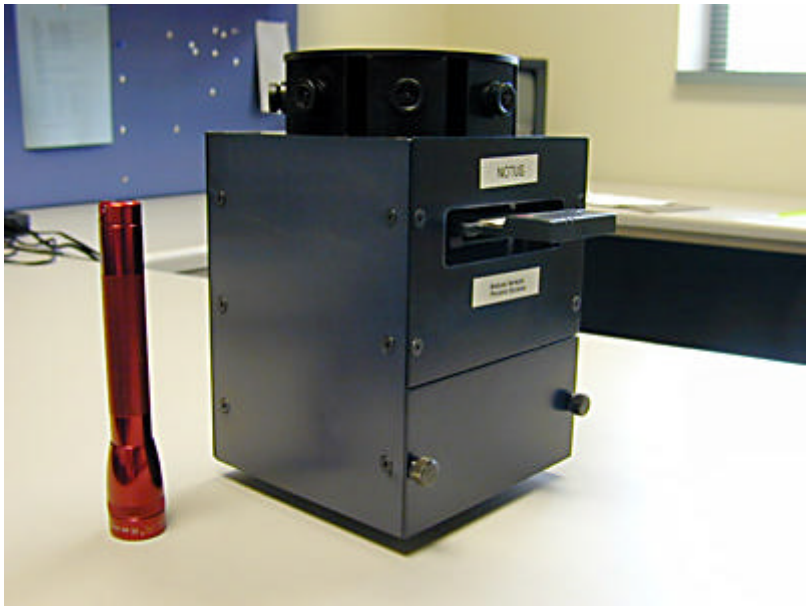


Figure 3. Packaged sensor module. 4 CMOS cameras are distributed across the top of the module.

The sensor modules are powered by either an AC adapter or a set of NiMH batteries. The batteries provide a means for field testing the sensor array although for only a short duration. Since the current set of modules was designed for processing flexibility as opposed to power efficiency, the current battery lifetime is approximately one to two hours. However, advances in microprocessor technology (e.g. the Carusoe chip from Transmeta) and upgrading the software application to throttle module operation until demanded will improve battery lifetime from two to ten-fold.

System functionality is distributed over several sensor modules and control and display platforms. The software architecture is also divided into several components operating across the network. The sensor module runs a reduced a reduced set of the Linux operating system. Linux was selected because it provides maximum flexibility in developing sensor drivers and code and is relatively simple to reconfigure. A web server operates on each module providing a means of serving data and images and also, through the cgi-bin interface, provides a portal for receiving control information and queries. By using Internet web protocols, the module takes advantage of many internet security features for restricting access.

The initial primary image acquisition and processing code is a C application that provides connectivity via software sockets. Using standard TCP/IP networking protocols, the application carries out commands to acquire and transmit images, perform background subtractions, stitch together camera images to create a panoramic view, compress images, and perform other assorted image processing activities. Each new data connection forks a new process thread, thereby providing the module with the ability to serve multiple requests.

Data and network security are fundamental issues for restricting access to this intelligence gathering network. The wireless networking cards selected for this application provide 128-bit encryption to the digital data stream. In addition, module access is restricted in the application by posting a password request/challenge whenever a new module socket is initialized.

The current control and display interface currently operates in either a laptop or desktop computer environment. However, our design criteria includes the opportunity for creating interfaces for wireless personal digital assistants and other display systems that are more appropriate for mobile law enforcement agents. Keeping with our open architecture philosophy, we have designed our control and display interface using Java. By using Java, the control and display interfaces can be embedded in the familiar web browser paradigm or can be operated as independent applications.

Using the control interface it is possible to control and display multiple video streams. The brightness, contrast, and frame rate can be remotely adjusted. A panoramic view from an appropriately configured module can be displayed or multiple video streams from multiple modules can be displayed simultaneously. Currently a rudimentary tracking function is implemented, with plans for a larger scale coarse 3D environmental reconstruction currently under development.

3. SENSOR NETWORK FUNCTIONS

The current demonstration system is comprised of as many as six wireless modules (each equipped with a multiple video camera sensor head), a module equipped with an RSI sensor, one or more laptop control and display systems, and a wireless network access point. The wireless network access point provides a link to a wired Ethernet network and is not a crucial part of the demonstration.

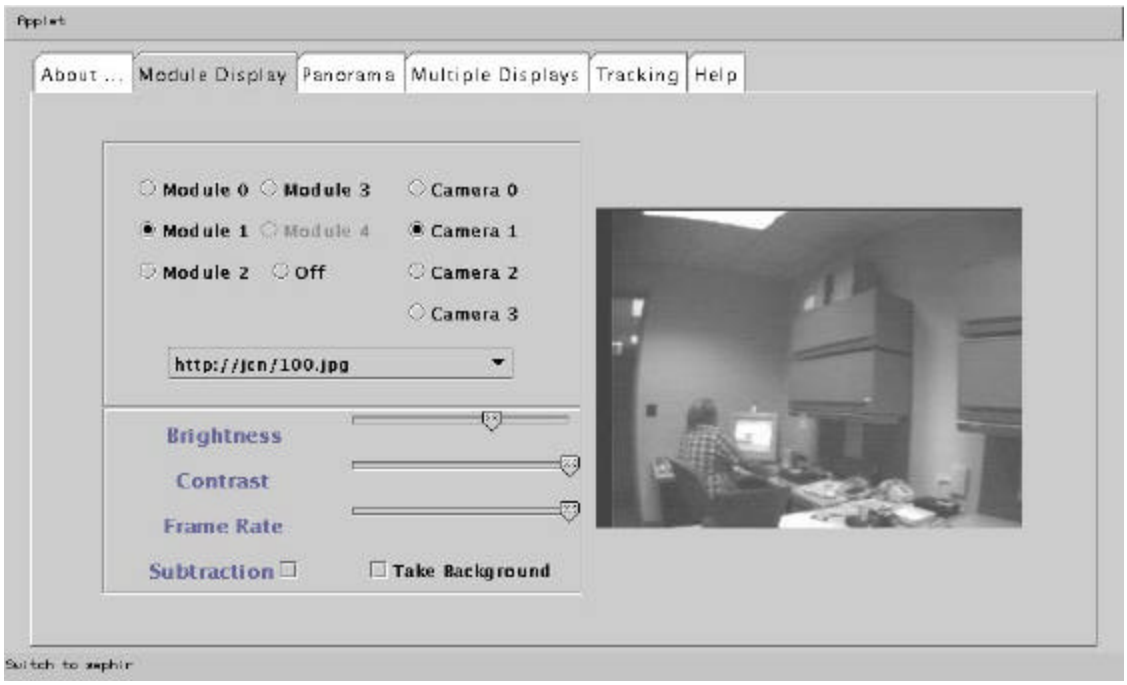


Figure 4. Screen shot of the Java interface showing module and camera selection and video frame.

Before any control and display software application is allowed to connect with the sensor module, the exchange of a username and password tokens must be satisfied. This security feature restricts sensor field usage to qualified personnel. In the current implementation, a security screen prompts the user for this information. We envisage that future deployment

would include control and display devices with embedded security tokens that are not easily accessible for theft and could be swiftly disabled if the unit is not under the auspices of qualified agents.

The control and display application is designed to operate within a heterogeneous sensor environment. As the user enters the sensor space, each module registers its availability and potentially alerts the user to hazards or atypical events. The goal is to present information in a manner that is quickly and effectively accessible to the user. Thus each sensor information panel is customized. For example, the video sensor panel shown in Figure 4 presents a video display updating at between five to ten frames per second. Various similar modules and cameras can be easily selected using the graphical interface. Also shown are controls that remotely tune the brightness and contrast of the camera. There are additional image processing filters that may be selected. In this example, the user may acquire a background reference image. When this reference is subtracted from subsequent frames, the appearance of motion is easily detected. In these displays, the images are transmitted between module and display system in a compressed JPEG standard format. This data flow could be greatly reduced if only motion sensing alerts or feature discoveries were reported. This interplay between the user and the wealth of sensor information will be an ongoing study for us as we create ever more powerful sensor fusion systems and interfaces.

The 180-degree panoramic view feature is illustrated by the screen snapshot shown in Figure 5. Each panoramic image is generated at the module from 4 video frames captured sequentially from the four cameras. In this case, the overlapping areas of the frames are removed and the remaining pieces are translated and stitched to form a continuous image. This simple procedure does not fully eliminate image distortion produced by the inexpensive, wide-angle lenses; however, the process is computationally efficient and generates a highly informative video stream.

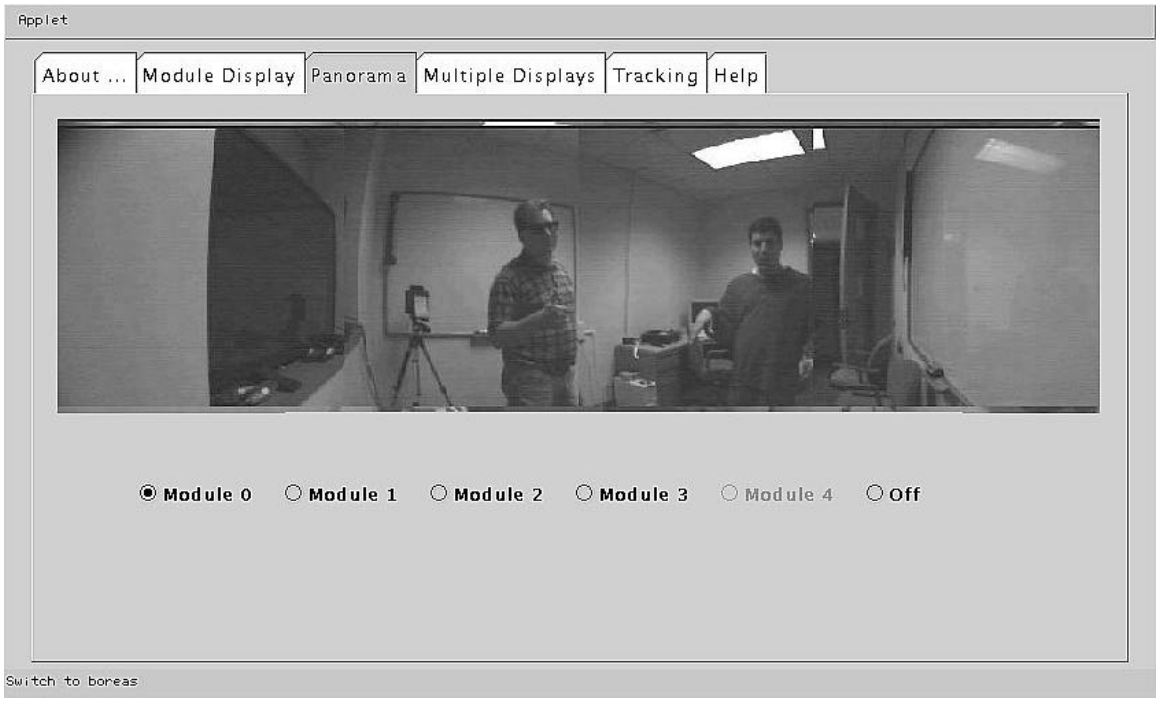


Figure 5. Screen shot of the panoramic view with module selection controls.

The multiple video demonstration demonstrates the power of using the wireless digital network architecture and is the prelude to further data fusion. The display presents video streams from several modules. It is feasible to design a system where the sensors prioritize and decide on the display presentation based on criteria applied to analyzed events. The user could then override this categorization once they determined which features were most critical to their mission.

Finally, a rudimentary motion sensor and tracking function is shown. The simple algorithm searches for the brightest image element, simulating how an infrared camera would sense a warm body. This bright pixel set is tracked by two sensor modules, triangulated and mapped onto an updating coordinate space on the interface. In this figure, the subject stood up, moved about the room and then returned to their desk. This application illustrates how what was once a high bandwidth stream from multiple video and assorted sensors has been reduced to a much low bandwidth detection and tracking stream that has much higher relevant information utility for the user. The inclusion of more modules will also provide three-dimensional reconstruction of the environment, similar to the Argus sensor space, only on a coarser level. This additional information will allow filtering based on height, size, and 3D trajectories which, together with feature selection based on color and even appearance, will serve to quickly identify target subjects while excluding distracting background events.

Eventually, more elaborate sensors can and will be added to the sensor space on a variety of scales. For example, it will be highly beneficial to add biomedical, tracking, and communication devices to an agent so that a remote unit might also monitor the health and safety of the agent. The agent might also activate and deploy additional networked sensors and/or robotic units to provide additional specialized information. Potentially services could also be provided that periodically collect sensor data in order to archive nominal events and thus create new filters that trigger on "out of the ordinary" or suspicious actions.

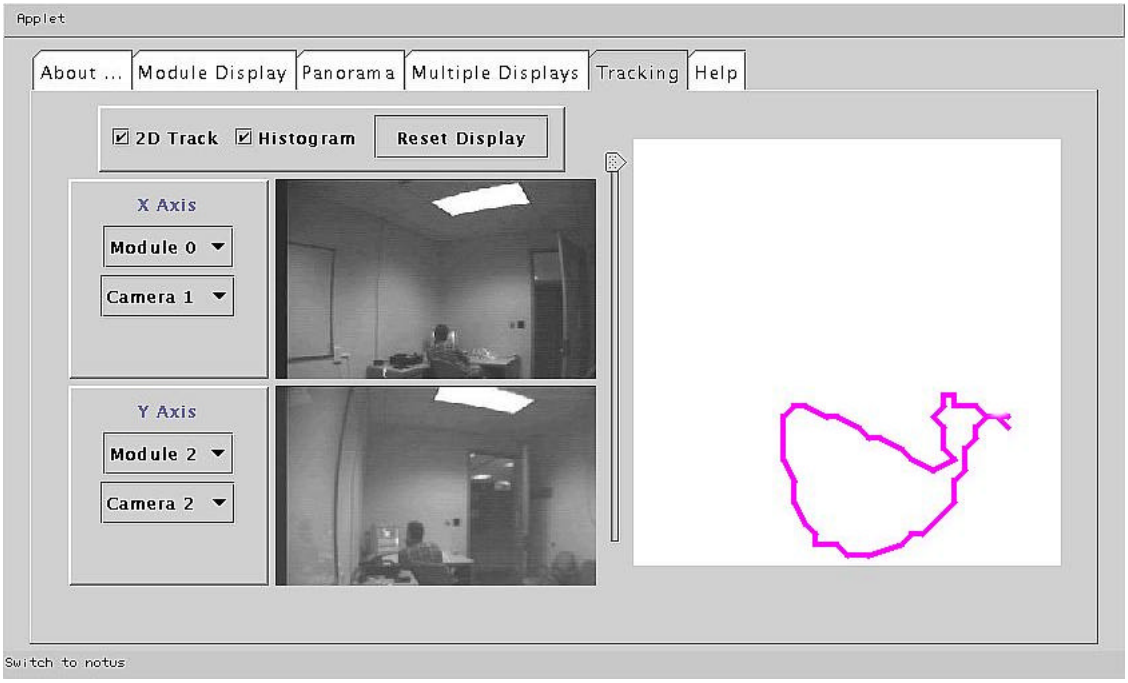


Figure 6. Screen shot of the object racking panel.

4. CONCLUSIONS

The integration of sensor and processing nodes collectively organized on a wireless network will provide many beneficial advantages to law enforcement officials and those who administer and manage these sites. Benefits of these systems will range from protecting the safety and welfare of agents entering a potentially dangerous scene, to the reduction of false alerts through the integration of heterogeneous sensors and the identification of potential threats or events that might ordinarily be missed. It is clear that the swift pace of technological advancements will lead to a revolution in site security management. And since cost projections should be tightly coupled with the constantly plummeting price of electronic components, it is likely that the initial cost of installation will eventually be the crucial factor in determining the degree of sensor deployment.

5. ACKNOWLEDGEMENT

This work was supported through DARPA's Tactical Sensor Program via Army Research Office Contract DAAD19-00-C-0099. The technical point of contact for this DARPA program is Dr. Edward Carapezza.

5. REFERENCES

1. K. Dixon, J. Dolan,W. Huang, C. Paredis, P. Khosla, "RAVE: A Real and Virtual Environment for Multiple Mobile Robot Systems," in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'99), Kyongju, Korea, October 17-21, 1999.
2. J. M. Kahn, R. H. Katz and K. S. J. Pister, "Mobile Networking for Smart Dust", ACM/IEEE Intl. Conf. on Mobile Computing and Networking (MobiCom 99), Seattle, WA, August 17-19, 1999.
3. M. G. Corr and C. Okino. "A Study of Distributed Smart Sensor Networks,". Dartmouth College, Thayer School of Engineering Technical Report Preprint, March 2000.
4. D. Estrin, R. Govindan, J. Heidemann, and S. Kumar. "Next Century Challenges: Scalable Coordination in Sensor Networks." In Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking, pp. 263-270. Seattle, Washington, USA, ACM. August, 1999.
5. Joanna Kulik, Wendi Rabiner, Hari Balakrishnan , Proc. 5th ACM/IEEE Mobicom Conference, Seattle, WA, August 1999
6. D. J. Brady, S. Feller, E. Cull, D. Kammeyer, L. Fernandez, R. Stack and R. Brady, "Information flow in streaming 3D video," SPIE Critical Review of Three-dimensional Video and Display, Photonics East 2000.

WIRELESS SENSING AND PROCESSING NETWORKS

BY

ANDREW MICHAEL RITTGERS

B.S., University of Cincinnati, 1998

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2001

Urbana, Illinois

DEDICATION

To my wife and family, who have lovingly supported me for many years. Thank you.

ACKNOWLEDGMENT

This work was supported through DARPA's Tactical Sensor Program via Army Research Office Contract DAAD19-00-C-0099. The technical point of contact for this DARPA program is Dr. Edward Carapezza.

TABLE OF CONTENTS

Page

1. CONTEXT OF THIS WORK

1.1 Thesis Overview

A set of smart sensors called the Medusa Network was developed by the Photonic Systems group at the University of Illinois for the purpose of developing a platform for tomographic analysis on distributed wireless ground sensor networks. The catalyst for creating the Medusa Network came largely from work done on the Argus Distributed Sensing and Processing Environment at the University of Illinois [1]. Argus is a Beowulf-class parallel computer designed as a test-bed for 3D imaging using distributed processing. The environment consists of a circular sensor space 4.3 m in diameter surrounded by 64 digital video cameras. These cameras are connected to the 32 dual-processor Linux machines that make up the Beowulf cluster. The cluster is capable of generating 128 x 128 x 128 cubic voxel sets at a rate of two frames per second, though improvements in software and camera hardware should speed the reconstruction rate up to eight frames per second.

A second catalyst of the Medusa Network was the desire to expand tomographic capabilities beyond the lab, to which Argus is confined. To do so, the major components of Argus would have to be made portable. This portable concept was experimented with JCN, the Photonic Systems group's mobile robot. JCN was a portable imaging platform based upon a robot designed by the ActivMedia Corporation. The robot contained a microcontroller that controlled basic low-level robot functionality and a separate onboard PC-104 computer, which provided high-level control, data processing, and human interaction. The computer communicated to the outside world via wireless LAN, through which users could login to the robot. Mounted atop the robot was a CMOS color camera and stepper motor assembly. Video was captured by the computer through a PC-104+ frame capture card. The motor was controlled through the computer's parallel port and could spin the camera 360 degrees. JCN proved to be a success of portable tomographic imaging, and eventually captured reasonable 3D volumes. However, JCN revealed several drawbacks of this type of setup. Processing time proved to be a major hurdle. Since JCN was limited to a single onboard computer, the process of reconstruction overwhelmed the single processor. Also, with a mobile platform, JCN lacked the ability to maintain strict alignment, inducing high noise levels into the system. Therefore, a desire arose to maintain the portability of JCN, but keep it on a stable platform. Rather than having one large sensor take the time to travel to multiple viewpoints and re-align itself, the group decided to

deploy multiple small sensors that are quickly aligned once at fixed locations. The latter setup also allows for the concurrent capture of data from multiple viewpoints for live object tracking.

A network capable of providing this type of platform should have modules that are wireless, robust, easy to deploy, and conservative in their use of power. According to the specification described above for use in tomographic imaging, the modules in the network should also be decentralized and self-organizing. Decentralization implies that each node should contain a reasonable amount of processing power and the capability of making its own decisions without user intervention. These decisions could include network communication protocols, geographical location or orientation, and could be based upon information gathered from its own sensors or from its peers. This decision-making ability makes the modules (and therefore the network) self-organizing by making the network dynamically scalable to include both large and small arrays.

The remaining sections in this chapter describe related research efforts and available hardware technologies for use in this effort. Chapters 2 and 3, respectively, overview the hardware and software designed and constructed for the first generation of wireless ground sensors. Chapter 4 characterizes the capabilities of the network and resources used to implement it. Chapter 5 describes the initial setup, deployment, and testing of the basic network as a tracking system. Chapters 6 and 7 discuss the future directions and conclusions drawn from the project.

1.2 Background

Surveying the battlefield, today's military leaders are constantly looking for new ways to measure their opponents. Today's battles underscore the fact that the value of information about the enemy has outpaced the value of strengthening one's own army. Today's victors are winning not because they had the better army, but because they knew more about their opponent than their opponent knew about them. This information gathered about the enemy can range from the enemy army's position to its strength and composition. Of particular interest to military leaders is the advancement of battlefield surveillance and modeling. New reconnaissance opportunities are unfolding with the advent of smart sensors linked via digital wireless networks. Such networks could be employed behind enemy lines to locate and track objects of interest throughout the battlefield.

Recently, the military has funded smart sensor related projects studying a variety of sensor issues. These include, but are not limited to, the logistics and deployment of sensors, sensor communication, sensor power, and sensor processing. Projects that have studied sensor communication have investigated the transmission frequencies and protocols that could be used by sensor networks. Projects that have studied devices have researched all types of devices from small fixed sensors the size of a grain of sand to large mobile platforms such as an all-terrain-vehicle. Airborne vehicles have also been studied that range in size from small airplanes to party balloons.

As an example of a communication project, the MITRE Corporation, with support from the Army, is working on a project to study the issues related to the security and detectability of short-range data links. As the Army deploys sensors wider and deeper into the battlefield, better control and accuracy are needed to prevent target misidentification and fratricide, or "friendly fire." The sensors must be better able to communicate accurate information among themselves and with a control center. The project searches for ways to implement "a flexible, common communications and sensor fusion architectural solution" for sensor systems [2]. Such a system must contain provisions that protect transmissions from discovery and exploitation by the enemy. Transmissions must also be immune to enemy disruptions that would incapacitate the network.

As an example of sensor device research, a project being studied at the University of California at Los Angeles researched the development of low-power wireless integrated microsensors (LWIM) [3]. The designs for these devices call for sensors the size of a cube of sugar, or about 1 cm^3. A related project is underway at the University of California at Berkeley to develop devices that act as smart dust, on the order of 1 mm^3 [4]. These sizes are achieved by using highly integrated MEMS technologies. Due to their extremely small size, these devices explore the limits of power conservation, power generation, and wireless communication. Although these projects spend a considerable amount of time ensuring that the incredible decrease in size does not negatively affect the capabilities of the sensor, the novel techniques used to make the sensors functional can inspire new methods for all sizes of networks.

In addition to battlefield modeling, uses of wireless sensing technologies can be expanded to include applications such as site security and monitoring. Information from the network can be made quickly and efficiently available to an authorized agent moving within the sensor space who has a need for quick and reliable information. The network could also be

programmed to keep this information secure from nonauthorized parties. This information is not limited to 2D or 3D visual fields, but using algorithms on each module could be a fusion of data from dense arrays of sensors ranging from visible and infrared cameras and detectors to seismic, acoustic, magnetic, smoke, toxin, and temperature sensors. A working example of just such a network was developed at the University of Udine, Italy, for use at locations such as railroad crossings and airport runways [5].

Wireless sensor nodes are predicted to become common in everyday life as the costs of computing power continue to plummet. They will be found in buildings, cars, and city infrastructures to enhance the lives we take for granted. Examples can already be found in pagers, cell phones, and wireless personal data assistants (PDAs) which continuously stream more and more information to us from satellites and radio towers inconspicuously located around the globe. Additionally, computing power continuously infiltrates our lives, from the ever shrinking PDA, to the recently announced wristwatch computer running the Linux operating system. Today's smart hearing aids, which "tune in" to the speaker of interest to the listener, contain more processing power than the largest mainframe computers only a few decades ago!

As the hardware of computing technology continues on its exponential growth, one must consider the system in its entirety. One cannot overlook the importance of the software developed to operate on this hardware. One of the more difficult tasks in building a networked array of ground sensors is that of developing algorithms to gather information from the distribution of sensors and then integrating that information for the purpose of tracking and target identification. Related efforts have studied network communication and data fusion. The algorithms used to accomplish the tasks of these systems are constantly analyzed to find improvements in system performance. In the example given that examines railway crossings, the University of Udine used techniques similar to computer vision methods. Key elements of the algorithm are image capture, background subtraction, object localization, tracking, and classification. Statistics are compiled from 2D image streams and compared against a predefined database of objects. Classifications can then be made that, combined with the state of the system, could raise operator alarms. The MITRE Corporation project used methods of triangulation to determine the location of the target. Based upon acoustic data, the sensors would transmit this raw data back to a central processing unit, whether this was a similar but more robust sensor, or a central location that contained large amounts of processing power. Realizing

that this approach required large amounts of intersensor communication, considerations were given to adapt new algorithms that required less transmission, although none was implemented or discussed in detail.

One of the issues found in these studies centers around network granularity [6,7]. Granularity refers to the sensing and processing capabilities of each node within the network. In a system with fine granularity, sensors communicate all the information they detect to a central processor. In a coarse-grained system, target classification is implemented at the sensor level and sensors communicate classification results. Thus the sensor array could be used to combine raw sensor signals into a global model before attempting target analysis (fine granularity data fusion) or the array could be used to combine locally produced target analyses into a global analysis (coarse granularity data fusion). The fine-grained approach requires substantial data transfer between sensor nodes. The coarse-grained approach requires substantial processing power and memory at the sensor nodes.

When implementing target analysis on a sensor network, a central design issue is determining exactly what level of granularity one should assign to the sensor and processor resources. To accurately choose the appropriate level of granularity, we must determine how to optimally distribute the computation among the sensors and a central processor. This determination will depend on the algorithms chosen to implement the target tracking.

Considering the algorithms listed above, most of the data collected by the sensors is relayed back to a central location for processing (fine granularity data fusion). All this sensor communication can be extremely taxing on the system both in terms of bandwidth and power required to communicate within the network. One solution to this problem would be for the sensors themselves to process a portion of the data locally, transmitting only a reduced set of output information. Such an implementation would require the use of an algorithm designed to coordinate the processing of information within sensors, and would be able to assimilate this data quickly and easily for the user. One exciting prospect is the use of tomographic techniques in order to create real-time 3D modeling and analysis of the environment that is immediately accessible to authorized users. This solution, developed by the Photonic Systems group at the University of Illinois, involves the use of tomography to create models of the scene that can be quickly assimilated on the client side of the network.

Similar to X-ray imaging of teeth regularly performed at the dentist's office, tomography involves the projection of rays of light across a 3D environment. The patterns captured from multiple locations and orientations can be reconstructed to form a model of objects within the space. Tomographic analysis allows targets to be analyzed in their native 3D or 4D spatio-spectral spaces and removes many of the ambiguities of conventional 2D analysis. The angular distance between sensors that surround the object space can be referred to as the angular range. As the angular range of the captured target data is increased, tomographic analysis becomes more effective. The angular range can be increased by tracking relative motion between the sensor and the target and by cooperative target analysis across a sensor network. While both approaches are worth considering, the Photonic Systems group focuses in particular on distributed tomographic analysis, image analysis, and target abstraction algorithms.

Tomographic algorithms provide a natural basis for mitigating the amount of information that is to be locally processed or transmitted to a central processor. As an example, one could implement a hierarchy of algorithms that form tomographic models on disparate data types (source intensity and target probability densities are example data types). In such a hierarchy, low-level algorithms would form local models based on measured signal intensities. Higher-level algorithms would form probability models based on local processors' target identification. The design question is ultimately reduced to "How should processing and communication be balanced at each level of the processing hierarchy?" The answer to this question depends on several factors. For example, sensor density is critical. On very sparse sensor arrays, the information received by each sensor is likely to be independent of the other sensors. In this scenario, a coarse-grained approach is suitable. However, as one improves the array resolution by increasing the sensor density, sensor information begins to overlap and common processing of array data becomes increasingly attractive. Therefore, fine to moderate granularity approaches that emphasize low-level communication are more efficient on dense arrays because they can be integrated into array hardware, thus reducing the need for general-purpose central processing.

A second critical design issue of sensor arrays is network structure. Traditionally, a central processor gathers information from sensor nodes and combines the information in an optimal or efficient manner. While it is clear that a network with a star topology (all sensors connected to a central processor) will be able to extract a maximum amount of information from the collected data, this approach has a number of serious drawbacks. As mentioned previously,

the aggregate communication necessary between the sensors and the central processor forms a bottleneck due to the limited amount of available bandwidth and interference in a wireless scenario. In addition to these required communications resources, the central processor must have sufficient computational resources to assimilate all of the collected data. An alternative to such a centralized network is a distributed network, in which integrated sensors and processors communicate as peers. Although the central processor approach is easier to program and conceptualize, it is also less robust against processor failure and requires significantly more power and processing capacity in a single location. Under the distributed approach, local processing may be included in the sensor design and the network topology can be designed to enable tomography and classification through iterative belief propagation of simple, locally computed, information. An appealing potential feature of distributed sensor networks is the elimination of the need for global communication. For example, in wireless ad-hoc networks, sensors could use power allocation to adjust their transmission power until links to a relatively small number of neighboring sensors are established. Such a scenario would mitigate both power consumption and multiple-access interference.

Considering the points made in the preceding paragraph, choosing the right topology for a particular application may be just as important as how one implements it. One common tomographic algorithm explored by the group uses convolution and backprojection [8]. The convolution step weights the output of each sensor based on its orientation to the spatial points of interest. The backprojection step sums the values weighted to produce a source density at each point. This exact approach can be used to combine target probability densities. In addition to the standard convolution and backprojection method, we choose to explore silhouette reconstruction as a less computationally expensive method of tomographic reconstruction. Silhouette reconstruction is a binary method where a sensor contributes a yes or no response as to whether an object is present at a specific location in the scene. This method can be used as a quick means to determine regions of interest before more costly reconstruction methods are used. Practical implementation of these tomographic algorithms within our sensor module array will use distributed processing and distributed control to achieve model reconstruction. Intermodule data communication will be dramatically reduced through efficient analysis and by limiting exchanges to hypothesis verification and/or resolution enhancement. The initial scenario employs up to four sensor modules, each equipped with four cameras, to monitor a scene. By

limiting the reconstruction volume to a coarse resolution, the system should achieve a throughput of a few models per second. Also, by embedding the control throughout the distributed network rather than within a centralized control station, wireless laptops and personal digital assistants (PDAs) can easily connect with the secure network and immediately request information and displays of the reconstructed environment. Continued enhancements will eventually lead to event triggers where the sensor modules will identify specific activity and request user intervention.

1.3 Review of Available Hardware Technology

1.3.1 Notebook computers

The most readily accessible example of portable processing is the notebook computer. Today's notebook computers meet ever higher standards of computational power. From formatting the simplest documents to rendering the latest 3D video game graphics, notebooks are pushed to higher standards of visualization, connectivity, portability, power conservation, and raw computing power. At the extremes, today's notebooks have CPUs that operate at frequencies near 1 GHz, and are sure to achieve even greater speeds. With this power, they can perform the calculations of yesterday's room-size mainframe computers in a package now less than 1 inch tall. Notebook manufacturers continue to adhere to the unwritten rule that computer processing power will double every two years. As an example, the WinBook Z1 850 MHz laptop was recently tested and shown to outperform the average 18-month-old 650 MHz laptop by a ratio of almost 2:1 [9, 10].

All of this computing power is not without cost. Notebooks can prove to be rather expensive, not only in the marketplace, but also in the limited amount of operational time that exists before the batteries of the device need to be recharged. Coupled with the fact that most notebooks come with many devices unnecessary to the development of sensor array nodes, power conservation presents a challenge not easily overcome by today's notebooks. The toughest notebook batteries last an average of 6 hours, while a typical notebook averages only 4 hours. Additionally, the significant size and weight of common notebook batteries cannot be overlooked. When considering their use in ground sensor networks, the desire for an alternative solution quickly develops into a need for something better.

1.3.2 Palm computing

Today's palm computers, or Personal Data Assistants (PDAs), have provided a host of solutions to the everyday consumer. These devices pack reasonable computing power into a small package that is convenient to use and requires fewer recharges than their notebook counterparts. PDAs are usually well integrated into a small package that is very portable. However, similar to laptops, they suffer from having too much functionality, unnecessary for ground sensors, that wastes precious power and space. They also usually suffer from a lack of connectivity to common accessory devices and from limited compatibility with other computing devices. As an example, the Compaq iPAQ is a high-end PDA on the market today. The iPAQ's core is a 206 MHz Intel StrongARM processor with 32 MB of onboard RAM [11]. A 320 x 240 pixel touch screen LCD display forms the main human interface, around which most of the circuitry of the device is neatly packaged. The PDA can display email, take notes, organize contact information, and even store and present multimedia clips. It includes a microphone to record sound clips and a speaker to play them back. For the everyday user, this device is truly a wonder to behold. Besides the inability to make coffee, this device can leave many with little more to want from a personal organizer. However, all this extra functionality, which makes the iPAQ and other similar devices so appealing to the general public is the main drawback to using PDAs in a sensor network.

1.3.3 Embedded computing

After seeing the advantages PDAs have to offer, one cannot help but consider the option of constructing a custom PDA for use as a sensor node. A simple means of executing this task is to consider the embedded computing market. Constructing an embedded PC allows the designer to customize options to retain desired components and discard those that are unnecessary to system operation. A number of platforms exist within the industry, providing designers with standards that keep the system consistent. This consistency eases the processes of connectivity and system integration. These industry standards also increase the availability of accessories that easily integrate into the system. Although most platforms end up being larger than the custom packaging of PDAs, most consider physical size to be an important issue and therefore include size specifications in the definitions of their standards. More about this issue is described in the following chapters.

2. SENSOR MODULE HARDWARE

A network consisting of between four and six prototype sensor modules was to be constructed at the University of Illinois and deployed on a trial basis for evaluating sensor array operation. Each node should contain at least one component each for sensing, processing, and wireless communication. Each node could be expected to operate within an ad hoc network and to cooperate in the communication and fusion of data collected by its own sensors, and that collected by the sensors of its peers. At the initial stages of the project, each module, or node, of this first network would be constructed using off-the-shelf commercially available components. Future stages will develop new generations of networks that will employ a custom module design. These custom designs should be expected to conform to higher standards of compactness and power conservation.

2.1 Sensor Module Design Choices

As any experienced engineer probably already knows, the greatest challenge in designing a system of this complexity is system integration. Verifying the consistency of hardware components, interconnectibility, availability of software drivers to operate the hardware, and the ability to develop application software to operate and control the system are all challenges which must be met just to have a working system. Once these solutions are found, they must be refined to convert a working solution into a viable one. Although the eventual goal of the project is to create just such an efficient and viable solution, the first generation of prototypes falls acceptably short. The modules are not the fastest, smallest, lightest, or most power-efficient devices in the world. However, they demonstrate the feasibility of a network that achieves the goals we defined earlier and leave the refinement to future generations.

Ultimately, the solutions to these problems become specific to the goals one is trying to achieve. For example, the goal of this project is to perform tomographic imaging on a network of wireless unattended ground sensor modules. Video and infrared cameras would be the primary sensors used to achieve these goals. If modules were constructed that were incapable of incorporating information from cameras into the system, the network would be useless. Therefore, to some extent, the sensors themselves become the predominant forces in determining the makeup of the modules. This section discusses the challenges facing module construction and the solutions that were found and chosen.

2.2 Compatibility of Hardware

One of the first steps toward completing the task of system integration is insuring the ability to physically connect the device together. This involves the selection of components that are inter-operable or that conform to the same standards. Due to the large selection of off-the-shelf components, this task does not have to be very difficult. However, when other factors of size, power, and accessories enter the picture, this bottom line is not always a given.

The selection of the appropriate system began at the core of the modules, the CPU platform. Due to our desire for wide availability and compatibility with other components of the system, only the most popular processor lines were investigated. The list of leading contenders included Intel's x86 and StrongARM, Motorola and IBM's PowerPC, and Transmeta's Crusoe processor families. These architectures have a wide range of available hardware and software support, easing the effort required to integrate the system together. At the time of this publication, Transmeta's Crusoe processor family was just beginning to hit the market with limited support and availability and was never a serious contender for inclusion into the modules. However, the ideas that the Crusoe family promotes, specifically those dealing with power management, make the Crusoe a wonderful example of where the future of these modules could lead.

Although the CPU is a good place to start, the importance of the system interconnections cannot be overlooked. For example, how does the CPU physically communicate to all the devices on the system? Traditionally, one thinks of a motherboard and chipsets to handle this duty. However, in the interests of size and power consumption, which will be discussed later in this section, even the smallest baby ATX[1] motherboard fails to be satisfactory. How does one then satisfy these concerns of size and power consumption and still maintain functionality? For answers to this question, we turned to the embedded computing industry, where a variety of platforms exist that attempt to solve these problems.

Today the embedded computing industry flourishes with a multitude of designs and choices to solve the problems listed above. In the interest of saving time and other resources

[1] ATX is a motherboard specification developed by the Intel Corporation for personal computers (PCs).

spent by searching for all possible alternatives, we chose to concentrate our research efforts on two promising platforms for embedded computing.

The first choice was the CardPC platform. The most exciting feature of the CardPC is that it condenses the basic components of a personal computer down to about the size of a standard credit card. While it is a bit thicker than most credit cards, the 3-inch length and 2-inch width make for an impressively small form factor. Drawbacks of the CardPC technology available at the time of this publication are that CardPCs are limited by the amount of computing power they can achieve and by the type of connections that can be made to such a small form factor. Unfortunately, one of the costs of shrinking a package is the loss in the amount of real estate available to the components of the system. In this case, the more modern and powerful chipsets are disqualified from use in these systems due to their larger size. Therefore, the system is limited to outdated technology that is defeated by competitor's products. In addition to being slower, these outdated chipsets do not benefit from recent hardware enhancements, such as advanced BIOS features and the addition of the USB port to most PC systems. A second cost of reducing the package size is the difficulty that arises in being able to physically connect to the device. For their part, the manufacturers of these devices use a connector dubbed "EASI" (Embedded All-in-one System Interface), a high-density 236-pin connector, but this connector can be hard to find and cumbersome to work with in the prototyping stage.

The second choice was the PC-104 platform. Although its physical size is not as exciting as that of the CardPC, the PC-104 platform holds its own in its traditional 3.6-inch by 3.8-inch form factor. The leading benefit of the PC-104 platform is derived from its wide acceptance in the embedded computing industry. The PC-104 platform provides a reliable standard in a resonably small platform that is neither too undefined nor too restricted. In this sense, the PC-104 platform provides a comfortable middle ground from which to work, allowing for easy prototyping. Another benefit of the platform is its method of integration, where component boards are easily stacked one on top of another. Contrary to traditional computing platforms where accessory boards are added in a direction perpendicular to the direction of the motherboard, PC-104 accessory boards stack in the same orientation as the main board. Rugged and reliable male/female headers replace the standard edge-card connectors of the PC. This limits the growth in size of the final system to one direction, which is an important benefit when prototyping a system. This way, the designer knows the approximate size of the device in two

dimensions before it is ever constructed, and the third dimension is controlled by the amount of features the designer wants the system to have.

The PC-104 platform was chosen because it provided a computing standard that conformed to our size requirements and could be rapidly developed with little modification to the software and accessory hardware. The PC-104 platform also provided the flexibility we desired to implement and modify the system for prototyping and future needs. An assortment of PC-104 accessory components are readily available, most of which are relatively computationally powerful and have power conservation abilities.

2.3 Frame Capture

As described above, the sensors themselves largely contribute to the design and makeup of the final module. For this project, the main sensors were video devices, either one of the selected CMOS video board cameras, or one of the two available infrared cameras that were owned by the group. Therefore, one of the main tasks of the module was to capture and process the data from the cameras. These actions needed to be accomplished with great speed to afford the network pseudo-real-time processing capabilities. This particular requirement became the largest hurdle we had to overcome in the design and construction of the modules.

Even with today's "advanced technology," capturing video streams at full rates[2] can be quite a challenge. As an example, consider our system: although it could be considered simple by many, it still presents a large amount of data. The typical images are 320 x 240 pixels in size, in an 8-bit grayscale format. Data rates are calculated according to the equation

320 x 240 x 8 x 30(fps) = 18,432,000 bits per second

where the full frame-rate of 30 frames per second is used. That is 18.4 megabits of data that must be captured, processed, and analyzed before the next 18.4 megabits arrive a second later. In order to allow enough time for the CPU to analyze this image data, it was advisable that we use a method of data capture and transfer that was significantly faster than this rate. This would also ensure that the precious bandwidth needed for video transmission would not be taken up by any overhead required by the transmission channel. In many transmission systems, overhead is introduced because a specific protocol must be followed to allow the devices on the system to communicate. This protocol usually attaches "headers" and "footers," pieces of data that include

[2] A full video rate is defined here as 30 frames per second.

information that identifies the source and destination of the data, and how the data fit into the overall message the device is trying to send. This information takes up space (bandwidth) on the communication channel and reduces the channel's overall data throughput.

The three major methods known by the author were considered to handle the task of capturing data from the cameras. The first was using a PCMCIA frame capture card. Since we knew that we would be using a PCMCIA carrier board to hold the wireless Ethernet card, it made sense to simply fill the second PCMCIA slot with a frame capture card. However, after some investigation, we discovered that current PCMCIA systems were basically another form of the legacy and relatively slow ISA[3] bus. While this bus may have been barely able to handle the data, it would be completely congested with the traffic given to it by the frame capture card. Furthermore, this card would have to compete for the now precious bandwidth with the wireless Ethernet card, which was vital to the system's success. We also discovered that the Activmedia Corporation, builders of the robot JCN, had tried this method and were disappointed with the results, especially in terms of the speed of data capture or frame-rate they were able to achieve.

The second option was to capture data digitally using the recently prominent Universal Serial Bus (USB) port. USB ports (under the USB 1.1 specification) operate at high speeds of 12 Mbps or lower speeds of 1.5 Mbps. Many USB cameras can transmit live video with 24-bit color with frame size 320 x 240 at full rates (30 frames per second) over this interface. This specification is comparable to the analog National Television Standard Committee (NTSC) video standard at a lower resolution. Adopted by the US in 1953, NTSC contains 525 interlaced[4] lines of horizontal resolution at 60 Hz [13]. Due to its analog nature, the true vertical resolution is undetermined, and depends solely on the display device. The digital option was appealing due to the fact that digital signals are more likely to preserve the integrity of data. Digital transmission is more robust against noise interference than analog signaling and does not depend upon the quality of an A/D (analog-to-digital) or D/A (digital-to-analog) converter. For example, in a typical frame capture system with a CMOS camera, the digital data from the CMOS camera is converted into an analog signal (NTSC). This signal is then sent to the frame capture card, where

[3] The Industry Standard Architecture (ISA) bus standard was developed for the original PCs by IBM operating 8 or 16 bit words at 16 MHz [12].

[4] Interlacing involves displaying odd lines followed by even lines for a given frame. This implementation reduces the true frame rate of NTSC to 30 Hz.

it is converted back from NTSC to a digital format, frame by frame. This process is done by D/A and A/D converters, the accuracy of which determines the quality of the resultant image. Depending upon this accuracy, information can be either lost or corrupted. Therefore, by keeping the image digital throughout the transfer, information is preserved accurately and the data is not needlessly converted to analog format, just to be converted back to digital. The digital transfer medium also presented another advantage: the ability to control the cameras themselves. Some USB cameras support a broad range of commands for enhanced control of imaging. While not universally true, most analog frame-capture systems do not have camera controls built-in. However, a large disadvantage to using USB cameras is the difficulty in locating the USB port itself. While it is becoming popular among desktop systems, at the time this project was in the design phase, USB was a feature that was hard to find among off-the-shelf embedded systems. Software support and drivers for the USB were also immature and hard to find.

The third and final option for image capture was to use a PC-104+ frame capture card. Like most analog frame capture systems (including the PCMCIA system mentioned above), this method also suffered from the D/A and A/D conversions, inducing some inaccuracy. However, unlike the PCMCIA system, this method had the advantage of using the faster PCI-style[5] PC-104+ bus. Faster than the ISA bus, this bus handles 32 bits of data at 33 MHz, or four times as fast. From our experience with JCN, we were able to locate a board that allowed four inputs to be multiplexed. Therefore, we were able to use four cameras and, through software control, choose the one that was most appropriate as the source. We also had the option of capturing images from all four in succession and morphing the images together into one panoramic image.

It was the final option (PC-104+ frame capture) that ended up being the most appealing to us. Although we would still be limited by the A/D conversions, the search for a suitable core module (CPU) would not be limited to only those with USB support, which were few and far between. The card gave us reasonable performance and the option of mounting four CMOS cameras in a 180-degree array, giving the modules panoramic capabilities.

[5] The PCI Local Bus originated at Intel as a method of interconnecting chips, expansion boards, and processor/memory subsystems. It operates on 32-bit words at speeds up to 133 MHz [14].

2.4 Device Size

As with most wireless and portable devices, the physical size of the device must be considered at some point in the design process. Although it is not universally the case, smaller devices generally have a number of advantages over their larger counterparts. Obviously, the smaller and lighter the device, the more portable it becomes. Smaller devices are usually easier to position or mount in a desired location and orientation. They also usually require fewer resources and materials, which can reduce production costs and lead time. Finally, they usually integrate better with other devices; they do not require as much space within the global system.

As mentioned above, the PC-104 standard was selected as the platform on which our modules were based. Size played an important role in the selection of this platform. The PC-104+ standard supports a standard size form factor of about 3.6 by 3.8 inches, or about the size of a 3.5-inch floppy disk. This standard includes the PC-104 (ISA type) and PC-104+ (PCI type) buses as well as the common hardware mount points on the board (through holes). The buses can be designed with "stackthrough" pins such that a socket exists on top of the board, and pins protrude out of the bottom of the board. An embedded computer can be created by stacking component boards together to form a system. Board connections are the result of simply pushing the pins of the upper board into the matching socket on the top of the other. When complete, we constructed a device measuring about 8 inches tall with an area 5 inches by 5 inches square. This is not revolutionary, but as mentioned before, it was good for the first generation prototype model.

2.5 Power Consumption

The next issue that entered the design process was power consumption. Being wireless meant that power must either be generated by the device to support itself, or that a power pack must travel with the device as part of the package. Since we did not want to labor on creating an energy source, the simplest method and our primary solution was to use a battery power pack enclosed within the modules. Aiding us in this decision was the reality of a variety of battery sources and suppliers that existed to help us in this task. Operating the modules entirely on batteries meant that the amount of power supplied to the device was finite. In order to keep each device of the network operational for as long as possible, serious deliberation was given to the

amount of power consumed by each component of the system under consideration. Also under consideration was the device's ability to perform operations to conserve power.

A large portion of constructing entire modules that conserve power involves identifying the individual components that consume the most power and finding ways to reduce their consumption. In our design, the largest consumers of power were the CPU board, where most of the functionality of the module lay, followed by the wireless Ethernet card, which required large amounts of power to transmit data via RF waves. Although the wireless card included support for a power saving mode, we were limited in the amount of conservation by the manufacturer's implementation of such savings. Therefore, we chose a card that had reasonable power savings given the transmission characteristics we desired in a wireless link. Power conservation involved simply setting a software switch to "on." We had more room to play in the selection of the CPU board. Models ranged in power consumption from 5 W to "hogs" that consumed more than 10 W. We chose a model on the lower end of that scale, consuming around 6 W. The power consumption of many boards was dictated by their functionality and factors like added features. For example, boards with video display driver chipsets consumed more power than those without video. By simply eliminating unnecessary functionality and extra chipsets, the amount of power could be reduced to only that amount required to perform the duties of the module. In this pursuit, we chose a model that satisfied our needs, yet included no extras that would waste precious power.

Another consideration in the selection of a CPU module was its capabilities to reduce its own power consumption as a function of need. For example, the BIOSs of many modern notebooks have capabilities that allow components of the system to power themselves down during periods of inactivity. This power saving feature has extended the life of many notebooks from around 1.5 hours to over 3.5 hours. These features have also been added to embedded machines, extending their lifetimes in the same manner. The CPU board we selected retained these power saving abilities.

2.6 Data Storage

While most portable devices, including laptop computers, use small hard drives to accomplish the task of data storage, this solution can be bulky and power consuming at best. Hard drives also pose the threat of failure due to moving parts, in cases of shock or fatigue. Most

modern hard drives are also much larger in data size than is needed for the modules, and excess data space translates into wasted materials, cost, and power. The CompactFlash modules alleviate these problems by virtue of being completely composed of solid state devices. Although small in terms of data size, their much reduced power consumption and physical size provide ample compensation. The CompactFlash interface also adds the feature of easy interchangeability. Similar to a PCMCIA card in operation, a simple swap of preprogrammed cards can quickly modify the functionality of the module.

2.7 Device Quality

Finally, once the above characteristics have been defined and satisfied, the device must prove to be reliable. Devices either must not fail, or must have a backup in place to perform the duties of the failed device. Software must include workarounds that allow the system to continue functioning even after a failure.

As photographed in Figure 1, the final prototype module consists of three component PC-104+ boards, with an additional smaller accessory board mounted on the top of the stack. The core of the module is a PC-104+ 266MHz Pentium processor board with 64 MB of onboard RAM. This board was purchased from the Ampro Corporation, their Coremodule P5e. The board has power saving capabilities similar to those found in laptops, such as throttling the clock speed of the CPU during idle periods. A PC-104+ frame capture card with four multiplexed input channels was used to acquire images from four CMOS cameras. This board was purchased from the Imagenation Corporation, the PXC200. Stitching together the images from the four cameras, the sensor modules have the added feature of being able to generate 180-degree panoramic views. Interfaces were also designed to allow inputs from the infrared cameras that were used in the first system field test. The third board, a PC-104 PCMCIA socket board, integrated IEEE 802.11b

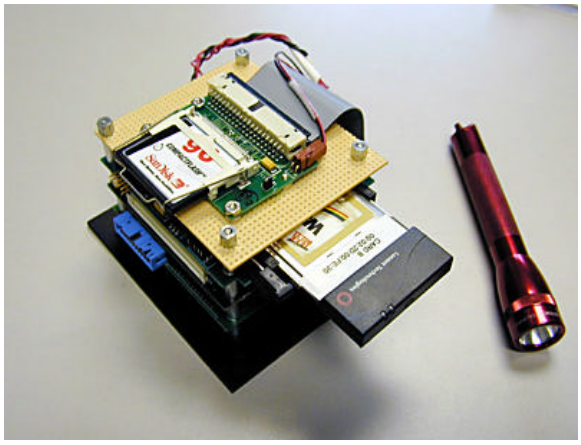


Figure 1. An exposed module is shown next to a 3-inch flashlight.

wireless Ethernet cards into the system. The wireless LAN cards were purchased from the Lucent Technologies Corporation as the WaveLAN IEEE Gold cards. They operate on a frequency of 2.4 GHz digital-spread-spectrum with 128-bit encryption for added security. They operate at a data rate of 11 Mbps, slightly faster than standard Ethernet, which operates at speeds of 10 Mbps. Drivers for these cards incorporate a standard power saving feature implemented by the manufacturer as mentioned above. The accessory board was comprised of a CompactFlash adapter card, and was smaller than the PC-104 cards. The adapter housed a 96 MB CompactFlash module that was used for data and system storage.

Each module was packaged in a custom designed anodized aluminum housing, which includes the 180-degree array of the four CMOS cameras and the power supply. Using eight rechargeable NiMH "C" sized batteries, each module can operate for more than an hour. By optimizing power management functions, that duration could be increased. Each module can also operate on an AC power supply for long-term development and testing purposes. As mentioned above, the prototype device was not designed for maximum power savings, and future custom designs should improve these values up to ten-fold.

Two additional modules were constructed and have differences from the above descriptions. One included a standard IDE hard drive in addition to a CompactFlash socket. It was used for the development of new modules and acted as a system backup for files. It was not used in any of the testing described below. The final module was the mini-RSI (Rotational Shearing Interferometer) module. This module contained a miniaturized version of an RSI and performed interferometric imaging. A little larger than the rest of the modules, it added the use of a PC-104 D/A analog output board and had only one camera. The camera captured the interference pattern generated by the mini-RSI for processing. The D/A analog output board connected to a piezoelectric element controlled the dithering of the path length of one arm of the interferometer.

3. SENSOR MODULE SOFTWARE

3.1 Operating System Selection

The largest software component of the system that needed to be built was the operating system (OS). Obviously, we did not have the resources to develop our own OS, so we needed to choose one from the set of those commonly available. From this list we chose the Linux operating system. Linux was chosen because it provides many advantages over other common systems. Perhaps the largest of these benefits is that Linux is a freely distributed open-source operating system released under a public license. Being free, Linux eliminated the monetary cost of adding it to our system. The open-source public license allowed us to fully customize the system and gave us total control of every aspect of its operation. In general, the open source concept and Linux have been welcomed by the general public, and Linux has become widely supported by many freelance open-source programmers. This network of support and resources enabled us to quickly integrate the system and make it operational.

3.2 Operating System Construction

A number of steps were taken to customize the system for network operation. First, using version 2.2.17, the Linux kernel (the core of the operating system) was reconfigured to maximize operational efficiency. Linux uses a kernel module loader, where instead of being hard-coded into the kernel, drivers not commonly used can be compiled separately from the kernel as modules. These modules can be dynamically loaded into the kernel when the system requires the use of such a device. During periods of inactivity, the modules can be unloaded to free up system resources. Only those functions essential to the system were compiled into the kernel and the rest were either compiled as modules for temporary use or completely removed. These exclusions streamlined the operation of the nodes by minimizing the amount of resources consumed by the kernel, leaving more available for the onboard applications. Component drivers not included in standard distributions were located from multiple sources and added to the system. For example, the latest WaveLAN drivers for Linux were downloaded from Lucent, configured, compiled, and installed. Software and drivers written by Rubini, et al. [15] for the Imagenation PC-104+ frame-grabber card were located on the Internet, configured, and even

edited for proper operation within our system. The software included functions that opened and closed the cameras for reading (grabbing images) and checking the status of data capture.

The file system was custom built for the modules. We began with a clean slate and built the system from the ground up. Based upon common distributions publicly available at the time, the directories commonly associated with Linux were created and populated with the files necessary to get the system operational. Files needed for basic system operation included common libraries, the modified kernel, and common configuration and settings files. After we felt the system was complete, we attempted to boot the development module. Initially, each attempt to boot the system failed for one or more reasons. After each failure, we would add the missing component that was needed to advance the boot process to the next step. A few reboots later, the system booted up completely. We still had a number of errors to eliminate from the system, but they were not of a critical nature. Over time, each error was individually removed to create a minimal error-free operating system. The final version of the complete operating system was reduced to less than 20 MB, running only those functions essential to the operation and stability of the network.

3.3 Application Software

Several server-side software applications were custom developed by the group and installed on the modules. Included in this list was an image processing and module control application dubbed “Imageserver” that provided connectivity through software sockets. Unlike Web servers that were designed to be primarily one-way communication channels, Imageserver was designed to provide two-way communications. Imageserver acted as a request handler, first authenticating a connection, then accepting requests. Example requests included image acquisition and transmission, background subtraction, panoramic view creation, and image compression. Replies from these requests could return objects ranging from a simple status update to an image stream. The modules could serve multiple requests by forking each process to a new thread for each connection. Imageserver also had the lua scripting language incorporated within its structure. This enabled the client to send scripts, or sets of instructions, to the server to be executed in quick succession. In addition to being freely available, lua provided an additional advantage by bringing a common interface to the network [16].

On the client side, a graphical user interface (GUI) was written in the Java programming language. The interface provided the user a number of unique viewpoints, displaying either live video streams or tracking data. For live video streams, the user could choose to view either a single camera from the network, or multiple cameras from one or more modules. The user could also view a panorama of images stitched together from a single module. With respect to tracking data, the user had a choice of which modules to accept data from, and could view reconstructed models of the object space. The user could choose how the model is displayed as one of three principle projections into the data space.

Basic Java applications benefit from being easily integrated with Web browser applications and relative platform independence, making Java an important factor for integration within a heterogeneous sensor network. While current implementations of the interface operate in desktop or laptop computer environments, the interface could be expanded to operate on devices such as personal data assistants and other handheld display devices.

4. DEVICE CHARACTERIZATION

4.1 Wireless Networking

The prototype modules use IEEE 802.11b PCMCIA wireless Ethernet cards produced by the Lucent Technologies Corporation. These cards are the WaveLAN Gold variety, and are capable of transmitting 11 Mbps on the 2.4 GHz frequency in a digital spread spectrum mode. The cards are capable of operating either in a centralized network mode, where all cards communicate through a base station to each other and the outside world, or in an ad hoc mode, where the cards can directly communicate with any other card within the RF range. The ranges of the cards vary depending upon the environment they are operating in. Office environments with many obstructions limit the range while open spaces allow for communications at relatively long distances. For example, a typical office environment may limit the range of the cards to 80 feet, whereas an open field will allow for communication at ranges up to 1/2 mile. These ranges can be extended with the use of omnidirectional antennas, which amplify the RF signal between the modules.

4.2 Power Requirements

In a full-on state, the prototype modules consume less than 10 W of power, consisting of about 7 W from the computer/processor core, 1.5 W from the Wavelan in peak transmission, and less than 1 W from the rest of the system. The 7 W of power consumed by the processor core at full speed translates into 16.1 nJ/instruction. Advanced power management techniques can be used which throttle the speed of the processor clock, reducing the number of instructions executed and thus the overall power consumption. However, even when the processor clock is temporarily stopped, the processor still consumes power. For example, when the processor is throttled back (clock is stopped) by 87.5%, power consumption only reduces by 70% to 2.25 W. This corresponds to a power consumption rate of 38.6 nJ/instruction, more than double that of the processor core at full speed. Therefore, a balance could be found in algorithms that group operations together to maximize the processor efficiency. For the sake of comparison, future use of processors such as the StrongARM could reduce core power consumption to 1 nJ/instruction. Considering the fact that ASICs typically outperform standard microprocessors by a factor of 100 in terms of power conservation, this value could be reduced even further to around 10 pJ/instruction.

Although the Wavelan card uses less power than the processor in power-saving mode, it is the greatest offender per bit of information. The 1.5 W of power used when transmitting at 11 MHz translates into 136 nJ/bit of information sent. Even when idle, the card drains a constant 50 mW from the power source. To maximize power savings, algorithms could be developed that power off the wireless card during known idle times, only polling it infrequently to gather systemwide updates. Relative to these two components, the power consumption of the individual pieces of the rest of the system was negligible.

Another loss of power is due to power conversion. In order to provide multiple voltages and conditioned power to the components of a system with a single power source, the power must be converted into the desired forms. This was done through the use of a DC-DC converter, which has its own limitations. Perhaps the largest drawback of its use is its efficiency, which hovers around 85%. Therefore, on a system that consumes 10 W, 1.5 W is lost and dissipated as heat in the DC-DC converter. Although more efficient converters exist, due to independent limitations, none is available off-the-shelf that is useful to this project. Future custom designs could incorporate a series of converters that benefit from the strengths of each.

4.3 Processing Power

Although the network is not capable of performing all operations at full data rates, the performance of this first generation is both acceptable and encouraging. Capturing raw data, the modules can stream video at about 10 frames per second. In panoramic mode, streaming four simultaneous images reduces the speed to about 2-3 frames per second. Using two modules, the network can track an object using triangulation methods with a refresh rate of about 4 Hz. Using silhouette tomography between three modules, a coarse 16 x 16 x 16 model is updated by the network at a rate of between 2 and 3 Hz. As network development and the evolution of software standards continue, these rates have continuously increased. The implementation of advanced image-capture techniques such as double buffering, etc., should increase video rates to around 20 frames per second. This enhancement would also increase the rates of panoramic streaming to around 5 frames per second. Although the simple triangulation tracking is no longer being pursued, planned software enhancements are expected to improve tomographic model rates to greater than 5 frames per second.

5. RESULTS

5.1 Tracking Field Test I

5.1.1 Experimental setup

A primary goal of the project is to demonstrate, using prototype modules, that tomographic data fusion is feasible using existing technology. To experiment with the newly constructed network, we outlined the first system field test. The scope of the test was to use tomographic analysis between two sensor modules to estimate the range and velocity of the specified target. Although the network had object detection capabilities, this test was not intended to be an experiment in object recognition. At the time of the test, the network provided both basic real-time analysis and systematic collection and storage of data for off-line processing. The latter was used to identify software enhancements and verify the accuracy of results. Future work on the network involves the automation of more of this processing. Limited by the site available, only 10 to 50 meter ranges were explored for testing. To maximize the accuracy of tracking, we used the IR cameras and a human subject in this test. The subject would easily stand out and ease the process of object identification due to a relatively large contrast between the subject and surrounding environment. Understanding our limitations on determining precise distances, as well as determining accurate camera orientation, we used a method of digital alignment, whereby we recorded a "reference" frame that captured the subject at an accurately measured location. From the data obtained from this reference frame, we were able to make estimates on the range of the object as it moved throughout the object space.

The test sequence involved two modules separated by about 5 meters. Data were collected for this distance as the subject moved in the sensor space on a preset course. The test was repeated with the separation between the cameras increased to about 25 meters. Full system testing would ideally examine system operations on fields with ranges upwards of 1 kilometer. However, due to site limitations only the shorter ranges have been explored to date. The movement of the subject was restricted as much as possible to one direction, perpendicular to the axis running through both cameras, intersecting that baseline at a distance halfway between the cameras. This restriction ensured that the test was repeatable, allowed us to make accurate measurements of the path, and limited the overall amount of error present in the setup of the test.

5.1.2 Processing and analysis

Using the algorithms described below, we were able to demonstrate object tracking on the 2D plane described by the locations of the cameras and the location of the subject. This was accomplished by a triangulation method similar to a restricted tomographic analysis. This method involved the identification of the object, the location of its centroid, and calculations to estimate its position in the field. Calculations were first conducted on the reference images, and subsequently on the data images. Data from subsequent images were then used to calculate an average velocity of the object. The information gained from the analysis served as a basis for future development of the network.

Data were collected in two steps, collection and off-line analysis. As we tested, we were able to learn a few things very quickly; we found that our biggest challenge was to accurately determine the location of the subject at all times for later verification. Our test was also hampered by bad weather — cold temperatures below 20 °F and more than 6 inches of snow — which tested the capabilities of equipment that, due to its prototypical nature, was not designed for all weather conditions. The movement of our subject was also restricted to a difficult trudge rather than a smooth walking motion. Despite these challenges, we were able to collect data for two cameras in the infrared spectrum at distances of 5, 10, and 25 yards using subject path 1 as shown in Figure 2. This completed the first step. Step two involved the task of processing and analyzing the data. The processing began by transferring the data collected on the nodes to a central location for ease of processing. In order to render an accurate representation of where a particular object is at a certain time, it is important that the modules and the data transmitted by the modules be synchronized in time. In order to facilitate this operation, timestamps were collected for each image and processed to verify data accuracy. This was done by declaring one of the two modules (the one on the origin) the reference module and the other the slave module. For each frame from the reference module, the algorithm searched for the frame of the slave module that best matched the reference frame with respect to time. In the first series of tests, the drift between modules varied from 0 to 4 frames per minute of video. To maintain accurate timing and tracking, the algorithm compensated for the slower module by dropping up to 4 frames per minute from the faster module.

With the images collected and synchronized, the analysis (object tracking) began. The first step in object tracking was to locate and identify an object within the images. A large

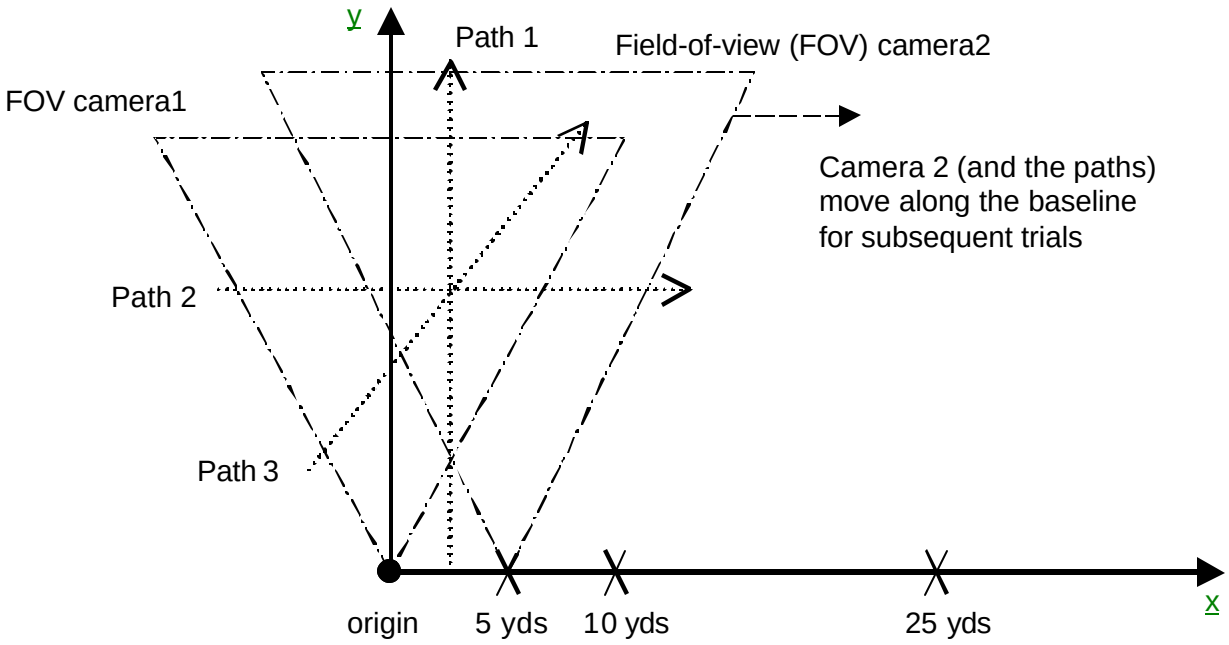


Figure 2. Diagram of test setup.

assortment of methods and literature exists covering object recognition and identification. We chose not to focus on this issue at this time, but rather to concentrate on the ability of the network to track the object identified within the images. We began by choosing a simple, automated identification method of object location based upon reasonable assumptions. The method involved thresholding the data images, where the grayscale image was first converted to binary. Those pixels above a certain threshold were highlighted, and the rest were combined into the background. The algorithm dynamically determined the value of the threshold based upon the size of the objects found to be above that value. Once an object was determined to be large enough to qualify and presumed not to be noise, the threshold was accepted and the algorithm proceeded. Adjacent pixels in the binary image were grouped into objects, and the objects were ordered according to size. The largest and brightest object was assumed to be the one we were most interested in, and was therefore chosen as "the" object. We then calculated the centroid of this object, and these coordinates were used to triangulate the position of the object on the field.

Although this thresholding method of object location worked moderately well in our tests, the most accurate method was to simply locate the objects by hand and have the computer find the centroid of the object we chose. This was a painstakingly slow procedure, as each frame

had to be checked by hand, but it ensured the accuracy of our analysis. Choosing the object by hand eliminated the negative effects of noise and system confusion. Noise was introduced as people walked through the rear of the test site, and can be seen in some of the images. Also, the thresholding algorithm tended to confuse the subject's head and legs; the switching back and forth between subsequent frames led to relative errors in tracking.

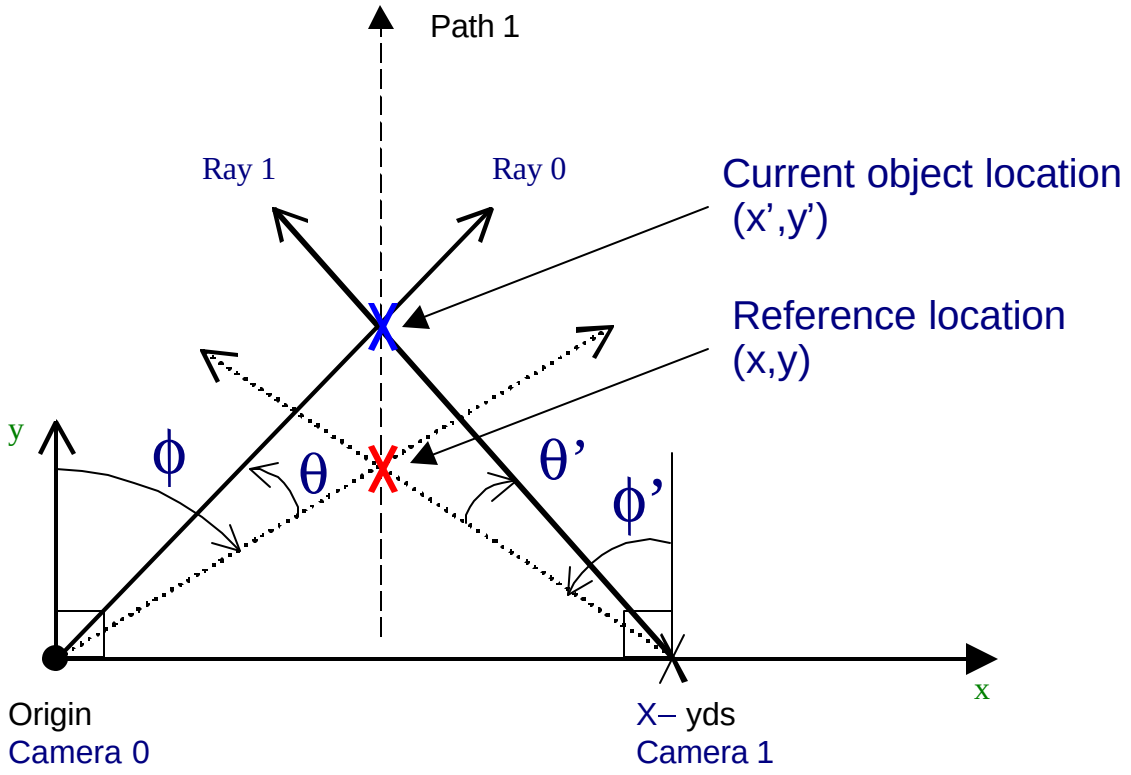


Figure 3. The angles as seen from above.

This centroid analysis was also used on the reference image pairs that were collected for each test. These centroid coordinates of the reference images, and those of each pair of data images, were used to triangulate the position of the object with respect to the cameras. This triangulation was completed as described by the equations and diagrams that follow.

The method of calculating the object location from the camera images is illustrated in Figure 3. This is accomplished by defining the line equations of the rays that travel between the object and the node. The parameters f and f' are calculated with the following equations:

$$f = \tan^{-1}\frac{x}{y}, f' = \tan^{-1}\frac{X-x}{y}$$

where (x, y) are the coordinates of the reference location, and X is the separation between the cameras. Each subsequent data image yields new values of q and q', thus revealing the location of the object relative to the cameras and reference frame. Values q and q' are equivalent to the ratio of the difference in pixels counted from the location of the object in the reference frame to the location of the object in the data frame to the overall size of the image, multiplied by the field-of-view of the camera. The values for these parameters are calculated with the following equations:

$$q = \frac{Px_0}{Ntotx} \times FOV(^\circ), q' = \frac{Px_1}{Ntotx} \times FOV(^\circ)$$

where Px represents the difference between the centroids of the current object location and the reference object location, ($Px = C_{obj} - C_{ref}$), $Ntotx$ is the width of the image in pixels, and FOV is the field-of-view of the camera in degrees. The slope of ray 0 (m_0) yields the ratio of the coordinates we are interested in (x', y'), which is dependent upon q and f. A similar calculation can be made on ray 1 (m_1) using q' and f'. Both slopes are calculated with the following equations:

$$m_0 = \frac{y'}{x'} = \frac{1}{\tan(f+q)}, m_1 = \frac{y'}{x'-X} = \frac{1}{\tan(f'+q')}$$

where values for f and q were calculated above, and X is still the separation between the cameras. Since the cameras are known to be on the lines describing rays 0 and 1, we use their locations to determine the y-intercepts of these lines (b_0 and b_1) according to the following equations:

$$b_0 = y_{cam0} - m_0 x_{cam0} = 0 \qquad b_1 = y_{cam1} - m_1 x_{cam1} = -m_1 X$$

where we define camera 0 to be at the origin, thus b_0 is zero, and the y values for both cameras are zero. We complete the range calculation by solving these simultaneous equations:

$$\begin{aligned} y' &= m_0 x' + b_0 \\ y' &= m_1 x' + b_1 \end{aligned} \quad \Rightarrow \quad \begin{bmatrix} y' \\ x' \end{bmatrix} = \begin{bmatrix} 1 & -m_0 \\ 1 & -m_1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \end{bmatrix}$$

to get (x', y'), the location of the object.

Results of the analysis, shown in Figure 4, are consistent with expectations described in the setup. The plots show the results of the two tests, one at a separation of 15 ft, and the other at a separation of 75 ft, as the subject walks almost directly away from the cameras at a distance

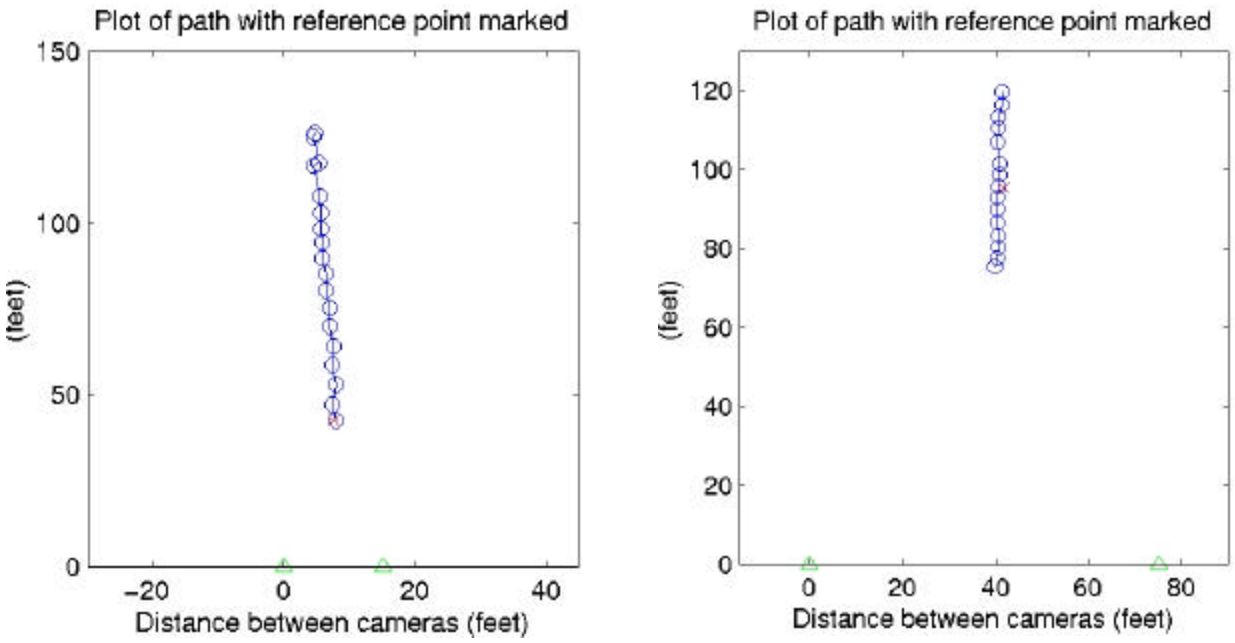


Figure 4. Plots of results from two trials of the system field test.

halfway between them. Circles represent the path of the subject, triangles represent the cameras, and the small "x" represents the location where the reference image was captured. The circles appear at regular intervals, which is consistent with the subject walking at a regular pace. The average velocity of the subject was calculated to be 2.71 feet/sec and 2.25 feet/sec for each of the tests. These values are similar to the average human walking speed of 1.3 meters per second, minus some resistance provided by snow [17].

Using the centroid method described above, we were able to demonstrate the network's ability to locate an object. Although this method proved to be reliable for our purposes, we desired to predict the amount of error present in our system. This error came from a number of sources, such as distortions of the imaging device, round-off error in our measurements, and other noise in the system. From our calculations, we found that the largest source of error was our measurements of the field-of-view of the cameras. Since this was an angular measurement, it proved to be significant only at longer ranges; the error was amplified as the range grew. For example, we calculated the error of our field-of-view measurement to be about 0.7 degrees. At shorter ranges around 40 feet, this 0.7-degree offset in the field-of-view caused less than an inch of error in the range calculation (y-direction). However, the same offset at ranges around 120

feet caused a 9-foot error in the range calculation. The lesson to be learned is that the accuracy of the knowledge we have about our own equipment is critical to the success of the network. Errors caused by other factors, such as resolution or distortion errors, were found to have an insignificant amount of error relative to the calculations mentioned above. An example of this would be a one-pixel miscalculation of the centroid. At short ranges of about 50 feet, an error of one pixel on one camera corresponded to a predicted error of 0.32 inches in the x-direction and 2.12 inches in the y-direction. An error of one pixel in both cameras corresponded to a predicted error of 0.65 inches in the x-direction and 0.01 inches in the y-direction. At longer ranges of about 100 feet, an error of one pixel in one camera corresponded to a predicted error of 0.80 inches in the x-direction and 8.59 inches in the y-direction. Errors of one pixel in both cameras corresponded to an error of 1.31 inches in the x-direction and 0.05 inches in the y-direction.

5.2 Tracking Field Test II

The third system test was performed using some software enhancements that provided real-time feedback on the tracking status. With a simple input of data collected from a reference measurement, we were able to track a bright object with relatively accurate results. For ease of object identification, we used a flashlight aimed at the cameras within the background of an office environment. The test was conducted on a short-range basis to accommodate the indoor facility. Again, performing two-dimensional object tracking, we set up the test with speed and simplicity in mind. The software acquired data by binning columns of pixels together. These bins were compared against one another in a winner-take-all fashion. The winning bin was the one that had the brightest overall value, and it was assumed that the object lay in this column. These coordinates were then entered into calculations similar to the triangulation method described above and range estimates were calculated. Results from this test are shown in Figure 5. It can be seen from this graph that the estimates conform to the ideal curve.

To begin the test, we measured a location in the center of the field-of-view of both cameras as well as the distance between the cameras. These exact (measured) coordinates were entered into the computer as reference parameters from which to base all future calculations. As the flashlights moved throughout the FOV of the cameras, the x and y coordinates on the 2D grid formed by the cameras were given by the computer.

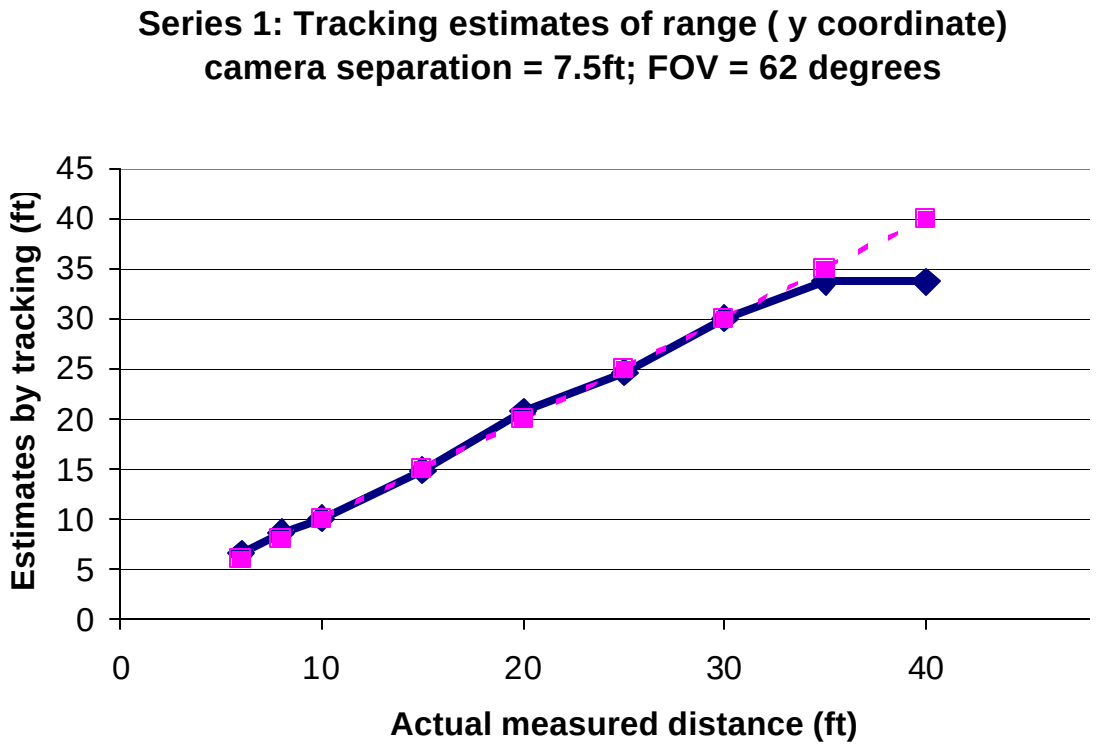


Figure 5. Results from system field test II.

A trade-off was found between resolution and processing time. Obviously, if the width of the bins is increased, there are fewer columns to process for the same size image, and calculations can be completed to update the position more frequently. However, the range resolution is sacrificed by not separating the columns into smaller discrete values. From our results, it appears that we may have run into this condition. As can be seen in the plot, the last couple of data points trail away from the ideal curve. This is caused by the finite number of bins, causing the system to lack enough bins to resolve the actual distance to the object. As the object travels evenly away, the system can only track it in discrete jumps that are only partially accurate.

6. POTENTIAL EXTENSIONS

As technologies continue to improve, they can be incorporated into the Medusa Network to produce smaller and more powerful generations of the system. The most radical improvements are likely to come from one of the six major areas of interest in the current sensors: processor platform, storage devices, wireless link, power sources, video capture, and focal planes.

As standard processing power in the marketplace (desktop computers) continues to grow, it should be no surprise to see the processing capabilities of embedded computers grow as well. However, unlike the exponential growth of power on the desktop, the embedded market has traditionally seen only a linear growth pattern. This is largely due to smaller market demand and the need for lower power units which require a longer design cycle. One recent development in this industry is the trend toward a system-on-a-chip. The system-on-a-chip concept means that what appears to be a processor is much more than a processor; it contains the equivalent of an entire motherboard minus the memory, graphics, and networking, in a single package. One example of this is the ZFLinux Corporation's MachZ system. Although some sacrifice is made in terms of processing power and speed, the power and real estate savings are worth consideration. Another exciting example is the Intel StrongARM processor family and its development. To date, this processor has led the market in power conservation, requiring only about 1 nJ per instruction executed. While the StrongARM is not as much of a single-chip solution as the MachZ, it outperforms the MachZ in certain applications while consuming less power. Both companies plan to announce the next generation of these processors within the year.

Currently, as much as 256 MB of RAM and 1 GB of CompactFlash storage space are available for use on an embedded system. These numbers are quite acceptable for the tasks currently being performed. As the size of the transistor on an integrated circuit continues to shrink, the storage size and power of computing devices should also continue their increase. In this way, the size and availability of both system memory and CompactFlash should improve without need for investigation of research projects similar to the author's. Therefore, as system tasks develop to require more resources, the size increase of the memory should be sufficient to maintain and improve the speed of the system.

Continuing the power discussion, the power sources are another area to consider for improvement. The largest-factor item to improve is the conversion process. A quick investigation into low-drop-out (LDO) linear converters reveals that in certain situations, they can be more

efficient than the switching converters used in the first generation of modules. However, they too have limitations relating to the variance of input battery voltage and amount of power they can handle. In practice, a complex system would need to be designed in order to further maximize the efficiency of the conversion process. A second item of interest in power sources is the use of Lithium-ion (Li-ion) batteries. Although they are more expensive, Li-ion batteries hold the highest energy density of any rechargeable battery available today. They are also lighter than the NiMH batteries used in the prototype generation, allowing for smaller and lighter modules for the same amount of run time. Additionally, recently developed supercapacitors could be employed to capture and store extra power for short-term use while maintaining battery charge cycles. Farther down the road of investigation into power sources, power generation could enter the mix of ideas. Possibilities have been conceptualized from various sources, including solar and chemical [3]. Although these technologies are still immature, their rate of development suggests they could be useful in the near future.

A third direction to pursue could be the wireless link. Although the IEEE 802.11b standard provides a rich set of features, it is not a perfect solution for ground sensor communication. Specifically, the standard lacks a reasonably long range [18,19] (near 1 mile is desired for long range communications on ground sensor networks). Also, a recent study completed at the University of Maryland suggested that the standard contains a number of significant security flaws [20]. Given today's available technologies, the standard could also provide higher data transmission rates. This is not to insult the designers of the specification, but to point out some of its inadequacies for ground sensors. Understandably, designing a communication standard such as the 802.11b is quite a challenge. Many advanced technologies and algorithms, from computer networking to power management to RF signal transmission, must be understood and employed to develop the system. This is why the humble author has chosen not to investigate this issue on his own, due to complexity and redundancy, since many large groups and projects are already dedicated to this study. Knowing this, future consideration should be given to the development and release of new standards. Just such an example is the high-speed IEEE 802.11a standard, which is soon to be released. This standard provides increased security, range, and data rates. For short-range communication, the recently released Bluetooth standard warrants investigation into its usefulness on ground sensors. Although not as powerful or as fast as IEEE 802.11b, the extreme low cost and power consumption are worth

serious consideration for short-range communications between sensors. Other projects have suggested the use of optical communications devices [3]. Although limited by line-of-sight communications, they could provide heightened security and tremendous power savings over traditional RF devices. Of course, to be useful, these technologies require some maturation of available hardware and software, although this may happen in the not too distant future.

As suggested earlier in this thesis, the use of digital frame-capture systems should be seriously considered for all future work in ground sensors. Although these systems were too immature for incorporation into the first generation, recent releases of hardware and software have fully incorporated USB devices into embedded systems. The use of USB has many potential advantages over traditional analog frame-capture systems. Included in these advantages are device control and advanced power management. Traditional analog frame-capture systems provide one-way communication between the camera and the capture device. USB has an advantage in being a two-way communication channel, allowing the computer to "talk-to" and control the camera. This can be useful in sending commands to the camera such as requesting higher or lower resolutions from the device. Traditional analog systems have no method of power management of the device, such as a camera. Usually, the camera and computer were powered by separate sources on separate cables, making control of the camera difficult at best. USB has another advantage in that it provides the power source for the device within the same cable, enabling the computer to control its use of power. This allows for tremendous power savings by empowering the computer to turn that device on and off at its discretion. Future generations of modules could investigate the continued development of USB2, which promises faster data rates for bandwidth-intensive operations such as video capture.

As a final consideration, focal plane technologies are continuously being developed for wireless sensors [21-25]. Research is being performed to develop uncooled microbolometer thermal imaging sensors, allowing the use of infrared cameras on ground sensors without the penalties of high power consumption normally associated with infrared arrays due to their need for cooling. High-resolution "smart" arrays are being developed that have "auto-zoom" capabilities similar to those seen on commercial digital cameras, but more powerful. These sensors allow the dynamic capture of both high- and low-resolution data based upon objects within the scene. For example, when an object of interest (such as a tank) enters the field-of-view of the device, the array can focus on the object at high resolution, while maintaining a low-

resolution capture of the background data. This allows the network make high-accuracy decisions about objects of interest while conserving precious bandwidth and power by not transmitting high-resolution information on areas that are mostly uninteresting to the user.

7. CONCLUSIONS

The success of the Medusa Network largely depends on the ability to find the balance between a centralized and a granular approach to distributed tomographic processing. It is the goal of this research project to determine this balance point for object tracking using our ground sensor network. Distributed tomographic analysis has many advantages over conventional imaging for target tracking using multiple sensors. By reconstructing targets in their native 3D environments, tomographic analysis yields information not available to conventional 2D systems. With this additional information, ambiguities normally present in the conventional system can be resolved to track objects more accurately. As such, we will continue to develop more sophisticated tomographic algorithms to explore the benefits of 3D and 4D analysis over conventional 2D tracking analysis. Future experiments involve tomographic methods of tracking multiple objects in a 3D volume. At first, this would be at very coarse resolution; however, as algorithms and communication protocols improve, tracking resolution will become more detailed with greater accuracy. In addition to improvements in algorithm design, our sensor network will be enhanced as small electronic devices continue to see incremental improvements in computational speed, power consumption, and component cost. A custom design using the technologies mentioned in the previous section could provide an ideal solution for a powerful and secure wireless sensor network that accurately analyzes and targets objects on enemy battlefields for months before losing power.

REFERENCES

[1] D. J. Brady, S. Feller, E. Cull, D. Kammeyer, L. Fernandez, R. Stack, and R. Brady, "Information flow in streaming 3D video," presented at SPIE Critical Review of Three-dimensional Video and Display, Photonics East, 2000.

[2] T. J. McAdams, T. L. Thomas, and R. Wade, "Communications system considerations for unattended army battlefield munition and sensor systems," in *MILCOM 97 Proceedings* vol. 2, November 1997, pp. 613-617.

[3] K. Bult et al., "Wireless integrated microsensors," in *Proceedings of Conference on Sensors and Systems (Sensors Expo)*, 1996, pp. 33-38.

[4] K. Pister, "On the limits and applications of MEMS sensor networks," Berkeley Sensor and Actuator Center, June 2001, http://bsac.eecs.berkeley.edu/~tparsons/PisterPublications/2001/DSSG_Pister.pdf.

[5] G. L. Foresti, "A real-time system for video surveillance of unattended outdoor environments," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 6, pp. 697-704, October 1998.

[6] M. G. Corr and C. Okino. "A study of distributed smart sensor networks," Dartmouth College, Thayer School of Engineering Technical Report Preprint, March 2000.

[7] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar. "Next century challenges: Scalable coordination in sensor networks," in *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking*, Seattle, Washington, USA, August 1999, pp. 263-270.

[8] L. A. Feldkamp, L.C. Davis, and J.W. Kress, "Practical cone-beam algorithm," *Journal of the Optical Society of America*, vol. 1, pp. 612-620, 1984.

[9] K. Kirkpatrick, "Speed thrills," *Mobile Computing & Communications*, vol. 12, no. 5, pp. 70-83, May 2001.

[10] R. Malloy, "Between a shock and a hard place," *Mobile Computing & Communications*, vol. 12, no. 4, pp.68-78, April 2001.

[11] Compaq Corporation, *Handhelds, iPAQ Pocket PC H3650*, April 2001, http://www.compaq.com/products/handhelds/pocketpc/h3650b.shtml.

[12] Techfest, “ISA bus technical summary,” 1999, http://www.techfest.com/hardware/bus/isa.htm.

[13] National Center for Supercomputing Applications (NCSA), University of Illinois at Urbana-Champaign, “Video standards, NTSC,” June 2001, http://archive.ncsa.uiuc.edu/SCMS/training/general/details/ntsc.html

[14] Techfest, “PCI local bus technical summary,” 1999, http://www.techfest.com/hardware/bus/pci.htm.

[15] A. Rubini et al., “Pxc 0.28,” July 2000, http://gnu.systemy.it/software/pxc/pxc.html.

[16] TeCGraf, “Lua: About,” 2001, http://www.tecgraf.puc-rio.br/lua/about.html.

[17] “Velocities (in meters per second),” class notes for Physics 7: Relativity and Cosmology, Department of Physics, University of California, Riverside, Fall 1998.

[18] Lucent Technologies Technical Staff, *User’s Guide for Orinoco PC Card*, Lucent Technologies, 2000, ftp://ftp.orinocowireless.com/pub/docs/ORINOCO/MANUALS/ug_pc.pdf.

[19] Lucent Technologies Technical Staff, *Sales Bulletin Performance*, Lucent Technologies, 1999, ftp://ftp.orinocowireless.com/pub/docs/IEEE /BULLETIN/SALES/sb_perfm.pdf.

[20] W. A. Arbaugh, N. Shankar, and Y. C. J. Wan, Department of Computer Science, University of Maryland, “Your 802.11 wireless network has no clothes,” March 2001, http://www.cs.umd.edu/~waa/wireless.pdf.

[21] B. D. Figler, “Uncooled microbolometer thermal imaging sensors for unattended ground sensor applications,” presented at SPIE Aerosense 2001, Orlando, Florida, USA, 2001.

[22] D. R. Frost and C. M. Durso, “High-resolution high-G 1280x1024 CMOS camera for use in unattended ground sensors and unmanned vehicles,” presented at SPIE Aerosense 2001, Orlando, Florida, USA, 2001.

[23] C. A. Kramer, D. J. Stack, T. H. McLoughlin, and S. L. Bogner, “Target detection and tracking experiment featuring a dynamically reconfigurable vision system fitted with immersive objects,” presented at SPIE Aerosense 2001, Orlando, Florida, USA, 2001.

[24] B. Pain, G. Yang, T. Cunningham, C. Sun, T. Jeddrey, J. B. Heynssens, B. R. Hancock, C. J. Wrigley, and R. C. Stirbl, “Low-power large-format imager family with smart image processing and wireless data link: status and prospects,” presented at SPIE Aerosense 2001, Orlando, Florida, USA, 2001.

[25] S. J. Ropson, T. R. Schimert, R. W. Gooch, P. McCardel, B. Ritchey and J. F. Brady III, “A Si 160x120 micro IR camera: Operational performance results,” presented at SPIE Aerosense 2001, Orlando, Florida, USA, 2001.

Report on
System Field Test I

Tomographic Imaging on Distributed Unattended Ground Sensor Arrays

Supported by:
DARPA - Tactical Sensor Program
Army Research Office Contract DAAD19-00-C-0099
Contact: Dr. Edward Carapezza

January 2001

Submitted by:
Photonic Systems Group
Beckman Institute for Advanced Science and Technology
The University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

Prepared by:
Andrew Rittgers

Abstract

The Medusa Network is a set of smart sensors developed by the Photonic Systems group at the University of Illinois for the purposes of testing object detection and tracking capabilities of tomographic imaging on distributed unattended ground sensor arrays. This paper describes the first system test performed in December of 2000.

Project Description

This paper describes the first system field test of the Medusa Network as suggested in the proposal "Milestones and Revised Budget of Tomographic Imaging on Distributed Unattended Ground Sensor Arrays" submitted to the Advanced Technology Office of DARPA. A network consisting of four prototype networked modules was constructed at the University of Illinois and deployed on a trial basis for testing of system capabilities. Each module, or node, of the network was based on off-the-shelf PC-104 computing components and contained the following:

- Pentium 266MHz processor board, with 64MB of on-board memory[1]
- Cards capable of frame capture at up to 30fps, with 4 multiplexed inputs
- IEEE 802.11b 11Mbps wireless ethernet card on PCMCIA board[2]
- 96MB Compact Flash drive (OS and data storage)
- 4 CMOS cameras (b/w - visible) arranged in a 180-degree array[3]
- battery pack and ac power supply[4]
- black anodized aluminum packaging

Two of the four modules had one of their four CMOS cameras replaced by a video-input port where the video feed from an infrared camera could be connected. These two modules were used in the infrared tests.

Each module was driven by a reduced version of the Linux operating system, less than 20MB in size, running only those functions essential to the operation and stability of the network. This version, or distribution, of Linux was developed at the University of Illinois by members of the Photonic Systems group and was based upon common distributions publicly available at the time. Several software applications installed on the modules were custom developed by the group. These include server-side image processing and module control applications, implemented in the C++ and Java programming languages. Additionally, a powerful Graphical User Interface (GUI) display application was written in Java for client-side use. The use of Java has become a cornerstone of the project due to its relative platform independence. Basic Java applications can be easily integrated with web browser applications, making Java an important factor for integration within a heterogeneous sensor processing system. As

---

[1] The processor board contains the newer PC-104+ capabilities, which consists of the PCI equivalent 32-bit bus. This allowed faster, full-frame data rate communication between the processor and frame capture card.

[2] The wireless ethernet cards utilize 128-bit encryption for increased security.

[3] With the lenses used, each camera has a field-of-view of approximately 60 degrees.

[4] The battery packs can operate the modules for about an hour. Since the test lasted much longer than this, ac power was the preferred supply.

Java becomes more widely available and implemented, both the simplest and most complex tools will be able to interact with the Medusa Network.

Two additional modules were constructed. One is currently used for development of new modules and backup files. It was not used in the testing described below. The final module is the mini-RSI (Rotational Shearing Interferometer) module, which adds the use of a D/A data acquisition board and has only one camera, which captures the interference pattern generated by the mini-RSI. The D/A board controls the dithering of the path length of one arm of the interferometer.

Experimental Goals

One of the main goals of the project is to demonstrate, using prototype modules, that tomographic data fusion is feasible using existing technology. Therefore, the scope of this system test was defined as follows:

Using tomographic analysis between 2-5 sensor modules estimate

- the number of targets
- the target cross section
- the target's range
- the target's velocity
- the target's trajectory

Some detailed notes follow:

1. The purpose of this system test was to experiment with the network's ability to track an object. The network will have object detection capabilities, but full-scale object recognition was not a consideration at this time.
2. At the time of the test, the network was capable of basic real-time analysis and also the collection and storage of data in a systematic manner for off-line post-processing. While this will be considered acceptable for this first test, future tests will incorporate the lessons learned from this test to have full real-time functionality.
3. Full system testing would ideally examine system operations on fields with ranges of 10m, 100m, and 1km. Due to significant weather conditions at the time of the test and the limited area the site provided, only the shorter ranges were explored.
4. In addition to using the CMOS cameras that operate in the visible spectrum, we characterized how well the system fared in the infrared region. Therefore, we used the two modified modules in the IR test. (Since we only have two IR cameras, we only used two modules in the execution of this test.)
5. Understanding our limitations on determining precise distances as well as determining accurate camera orientation, we used a method of digital alignment, whereby we recorded a "reference frame" that captured an object at a known and accurately measured location. From the data in this reference frame, we are able to make estimates on the range of the object as it moved throughout the object space.

Experimental Setup

The test sequence began with two modules situated relatively close together. Data was collected as a single object moved in the sensor space in known directions. The distance between the cameras was then increased and data was collected with the new separation. This cycle was repeated for multiple camera separations.

To conduct this test, we began by marking the field. A baseline was constructed of measured intervals between two cameras. Markers were placed at the designated origin, and at intervals of 5, 10, and 25 yards. For this two-camera test, one camera was placed over the origin, and the other centered over the appropriate interval length, beginning with 5 yards and increasing from there. In order to control the experiment and make it repeatable, we had the subject (the object was human) move in precise directions at known intervals. The movements were reduced to a single direction (such as the y-direction only) to find if there were particular weaknesses in one aspect of the tracking and not others. As an example, for the first test, the subject moved along a path perpendicular to the baseline at a distance halfway between the modules (in the y-direction). This movement tested the ranging of the network at a fixed trajectory. The second test was parallel to the baseline (in the x-direction), at a known (measured) distance. It was a test of the trajectory of the network at a fixed range. The third test movement was a combination of the two, where the subject moved at a diagonal across the space. The fourth was merely a random path to test tracking capabilities, a real world scenario. However, these last two tests reduced our ability to verify their accuracy as we don't know the true location of the subject at all points.

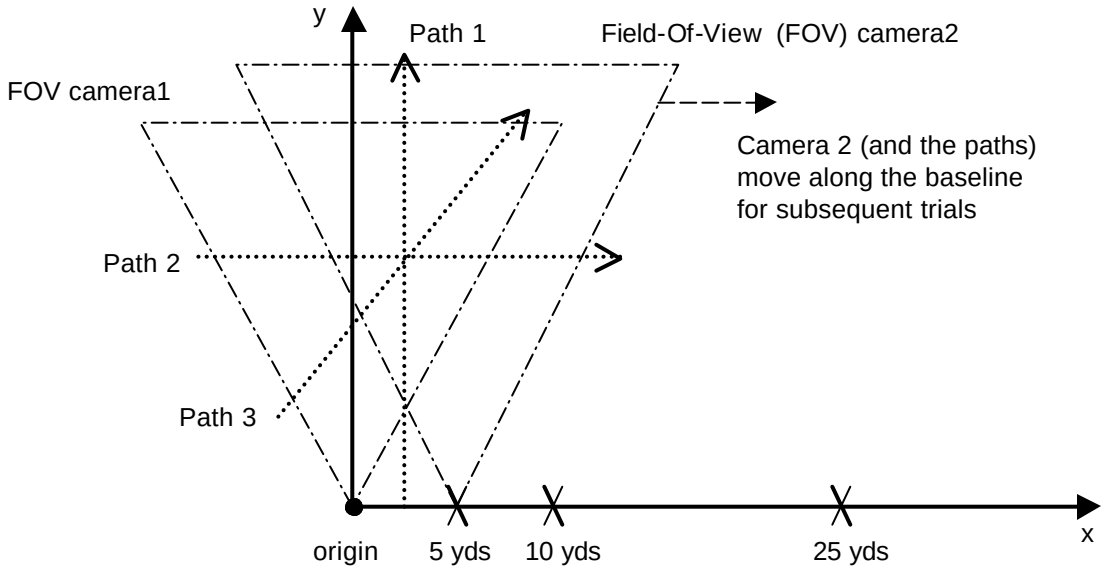


Figure 1. Diagram of System Test (not to scale)

Results

Using the algorithms described below, we were able to demonstrate object tracking on the 2-dimensional plane described by the locations of the cameras and the object. This was accomplished by a triangulation method, similar to a restricted tomographic analysis. Data was collected in two steps, collection and off-line analysis. The information gained from the analysis will serve as a basis for future development of the network.

As we tested, we were able to learn a few things very quickly; we found that our biggest challenge was accurately determining the location of the subject at all times for later verification. Our test was also hampered by bad weather, cold temperatures below 20 degrees F and more than 6 inches of snow tested the capabilities of equipment that, due to its prototypical nature, was not designed for all weather conditions. The movement of our subject was also restricted to a difficult trudge rather than a smooth walking motion. Despite these challenges, we were able to collect data for two cameras in the infrared spectrum at distances of 5, 10, and 25 yards. This completed the first step.

Step two involved the task of processing and analyzing the data. The processing began by transferring the data collected on the nodes to a central location for ease of processing. Next, the images were synchronized in time by using the timestamps collected for each image. This was done by declaring one of the two modules (the one on the origin) the reference module. For each frame from this module, the algorithm searched for the frame of the counterpart module that best matched the reference frame with respect to time.

With the images collected and synchronized, the analysis (object tracking) began. The first step in tracking was to declare and locate an object within the images. This could be done a myriad of ways. For automated tracking, we chose a method of thresholding, where the grayscale image is first converted to binary. This was done by highlighting those pixels above a certain threshold, and combining the rest into the background. The algorithm dynamically determined the threshold based upon the size of the objects found to be above a certain threshold. Once an object was determined to be large enough to qualify (presumed not to be noise), the threshold was accepted and the algorithm proceeded. Once the image was binary, adjacent pixels were grouped into objects, and the objects were ordered according to size. The largest (and brightest) object was assumed to be the one we were most interested in, and was therefore chosen as "the" object. We then calculated the centroid of this object, and these coordinates were used to triangulate the position of the object on the field.

Although this thresholding method of object location worked moderately well in our tests, the most accurate method was to simply locate the objects by hand and have the computer find the centroid of the object we chose. This was a painstakingly slow procedure, as each frame had to be checked by hand, but it insured the accuracy of our analysis. However, choosing the object by hand eliminated the negative effects of noise and system confusion. Noise was introduced as people walked through the rear of the test site, and can be seen in some of the images. Also, the thresholding algorithm tended to confuse the subject's head and legs; the switching back and forth between subsequent frames led to errors in tracking.

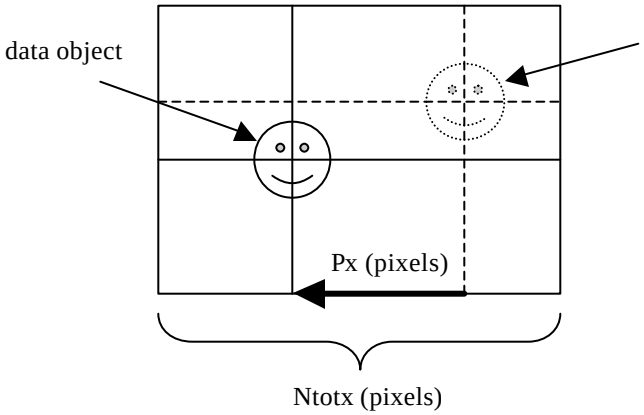
This centroid analysis was also used on the reference image pairs that were collected for each test. These centroid coordinates of the reference images, and those of each pair of data images were used to triangulate the position of the object with respect to

the cameras. This triangulation was completed as described by the equations and diagrams that follow.

Figure 2. Examples of typical images. The image on the left is an example of raw image data collected from one of the infrared cameras viewing the test subject. The image on the right is and example of a binary image, where the object (subject's head) has been selected and it's centroid marked.

Figure 3. Examples of abnormal images. The image on the left is an example of raw image data collected from one of the infrared cameras where the subject has almost faded into the background. The image on the right shows the noise induced when a person enters the scene in the distance. The figure is barely visible as tiny dots to the left of the subject.

Example calculations as seen from image frame :

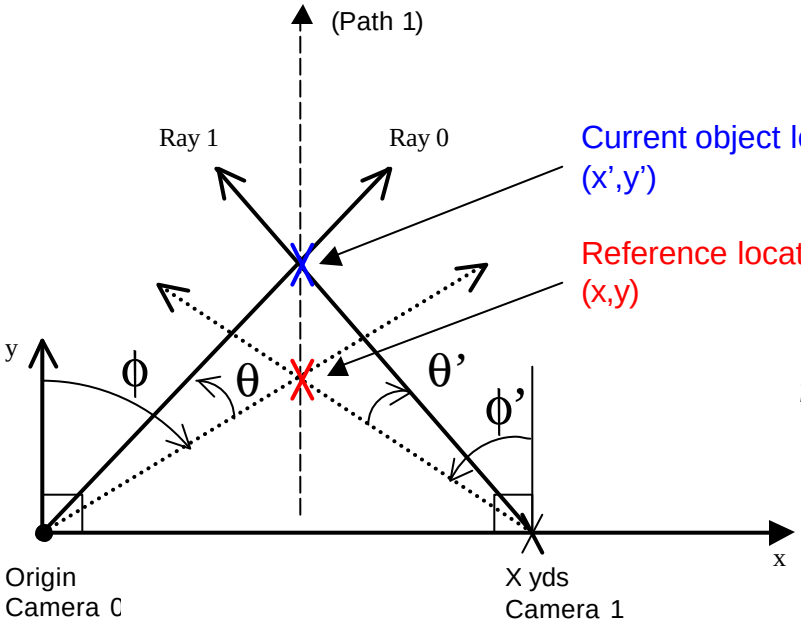


Cobj = Centroid of data object
Cref = Centroid of reference object

$$Px = Cobj - Cref$$

$$q = \frac{Px}{Ntotx} \times FOV(^\circ)$$

Example calculations as seen from above field:



$$\phi = \tan^{-1}\frac{x}{y}$$

$$m_0 = \frac{y'}{x'} = \frac{1}{\tan(\phi + \theta)}$$

Since the cameras are known to be on the line describing the rays, use their locations to determine the values of b:

$$b_0 = y_{cam0} - m_0 x_{cam0} \qquad b_1 = y_{cam1} - m_1 x_{cam1}$$

Solving these simultaneous equations yields the current location of the object:

$$\begin{aligned} y' &= m_0 x' + b_0 \\ y' &= m_1 x' + b_1 \end{aligned} \quad \Rightarrow \quad \begin{bmatrix} y' \\ x' \end{bmatrix} = \begin{bmatrix} 1 & -m_0 \\ 1 & -m_1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \end{bmatrix}$$

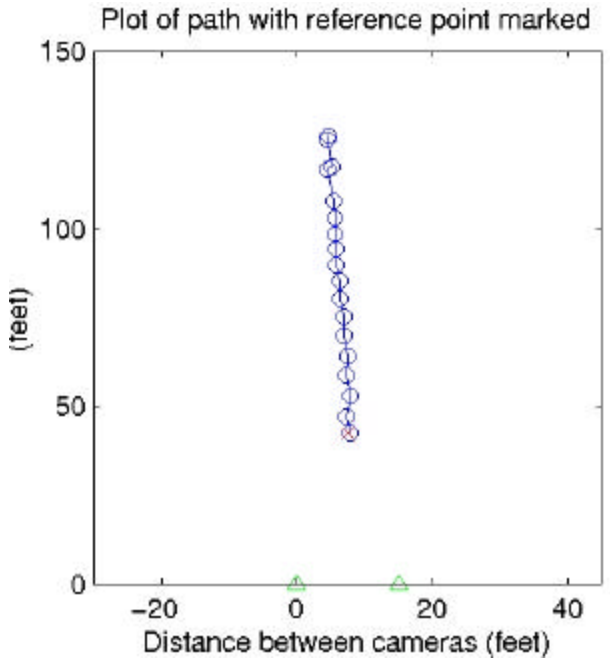


Figure 4. This figure shows the plot of the output from the tracking algorithms on a test with camera separation of 5m(15ft). The blue circles and connecting line represent the track of the subject calculated by the algorithm. The green triangles represent the locations of the cameras, and the red "x" denotes the location of the reference frame that was used to digitally align the system. It can be seen that the plot is consistent with the movement of Path 1, as the subject walks directly away from the cameras at a distance halfway between the cameras. The circles also appear at regular intervals, which is consistent with the subject walking at a regular pace. Missing circles are from frames that were skipped due to inadequate data.

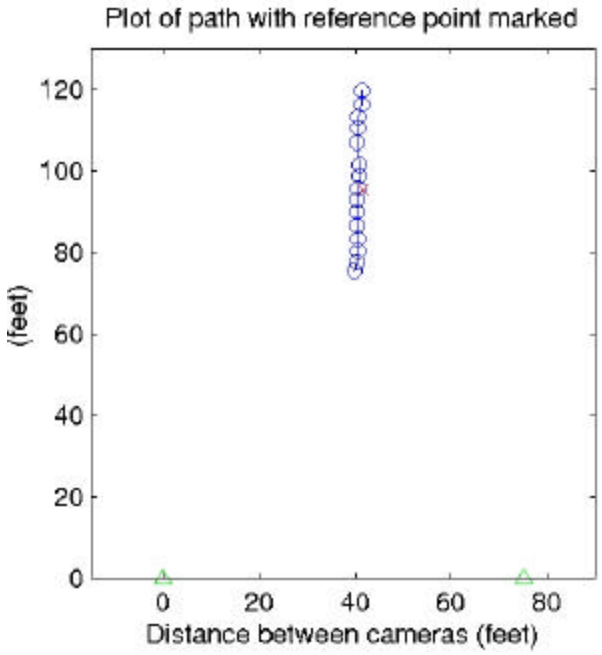


Figure 5. Again, another plot of the output of the algorithms; this time the cameras were separated by about 25m(75ft). This plot is also consistent with Path 1, which was completed at a longer range.

Using the centroid method described above, we were able to demonstrate the network's ability to locate an object. Although this method proved to be reliable for our purposes, we desired to predict the amount of error present in our system. This error came from a number of sources, such as distortions of the imaging device, roundoff error in our measurements, and other noise in the system. From our calculations, we found that the largest source of error was our measurements of the field-of-view of the cameras. This proved to be significant only at longer ranges, and expanded as the range grew. For example, we calculated the error of our field-of-view measurement to be about 0.7 degrees. At shorter ranges around 40 feet, this 0.7 degree offset in the field-of-view caused less than an inch of error in the range calculation(y-direction). However, the same offset at ranges around 120 feet caused a 9-foot error in the range calculation. The lesson to be learned is that the accuracy of the knowledge we have about our own equipment is critical to the success of our algorithms.

Errors caused by other factors, such as those with higher resolutions caused little or no error. An example of this would be a one-pixel miscalculation of the centroid. At short ranges of about 50 feet, an error of one pixel on one camera corresponded to a predicted error of 0.32 inches in the x-direction and 2.12 inches in the y-direction. An error of one pixel in both cameras corresponded to a predicted error of 0.65 inches in the x-direction and 0.01 inches in the y-direction. At longer ranges of about 100 feet, an error of one pixel in one camera corresponded to a predicted error of 0.80 inches in the x-direction and 8.59 inches in the y-direction. Errors of one pixel in both cameras corresponded to an error of 1.31 inches in the x-direction and 0.05 inches in the y-direction.

On the second day of testing, we completed some software enhancements that gave us some real time feedback on the tracking status. With some simple input of data from a reference location, we were able to track a bright object (here, a flashlight) with surprisingly accurate results. The test was conducted on a short-range basis to accommodate the indoor facility we were restricted to at the time. We set up the test with speed and simplicity in mind; here are some notes on the details:

- Using our original tracking software, we attempted to track the brightest objects located in the field-of-view of our CMOS video cameras
- The software used "bin-ing" where the pixels were grouped into columns and compared in a winner-take-all fashion. The winning column was the column that had the brightest pixel value, and it was assumed that the flashlight lay in this column. The column values were then entered into the calculations and range measurements were made, based upon the triangulation process previously described.
- To begin the test, we measured a location in the center of the field-of-view of both cameras as well as the distance between the cameras. These exact (measured) coordinates were entered into the computer as reference parameters from which to base all future calculations. As the flashlights moved throughout the FOV of the cameras, the x and y coordinates on the 2D grid formed by the cameras were given by the computer. Sample data sets from these coordinate values are plotted in the graphs that follow.
- A tradeoff lies between resolution and processing time. Obviously, if the width of the bins is increased, there are fewer columns to process for the same size image, and you can calculate more frequently. However, the range resolution is sacrificed by not separating the columns into smaller discrete values. From our results, it appears that we may have run into this condition.

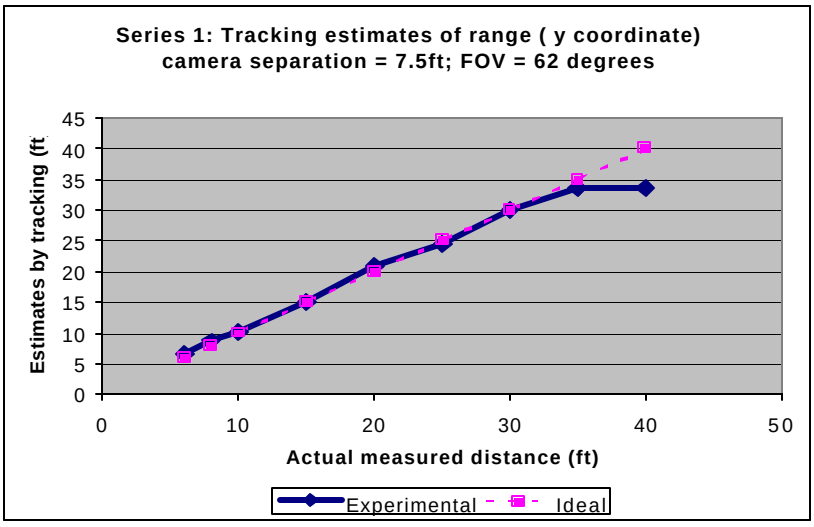


Figure 6. Experimental and ideal data sets of the range from cameras. The estimates given by the software were fairly accurate up to a range of 35 feet. However, at ranges greater than 35 feet, the estimates become inaccurate due to the low resolution induced by having bins that are too wide. We seek to reduce this error in the future by reducing the size of the bins without trading off performance.

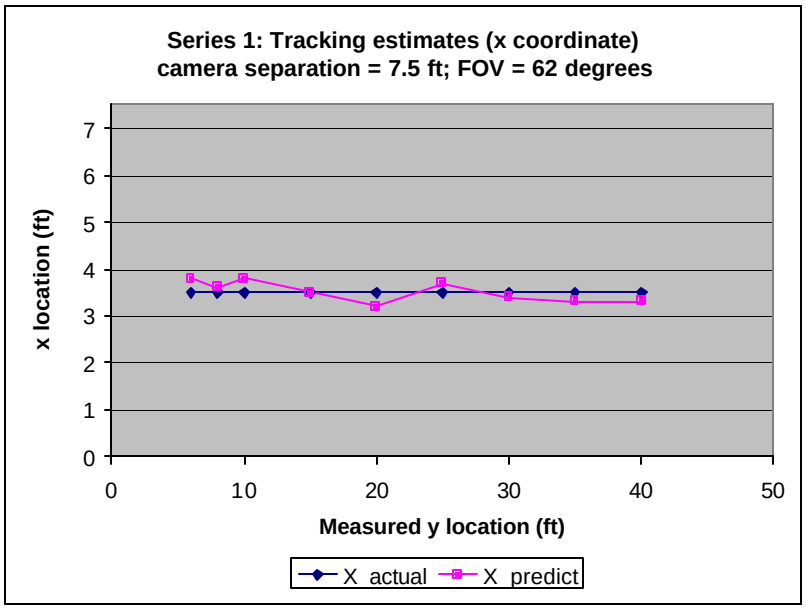


Figure 7. Experimental and ideal data sets of x coordinate versus the y coordinate shows that the x coordinate estimation is accurate

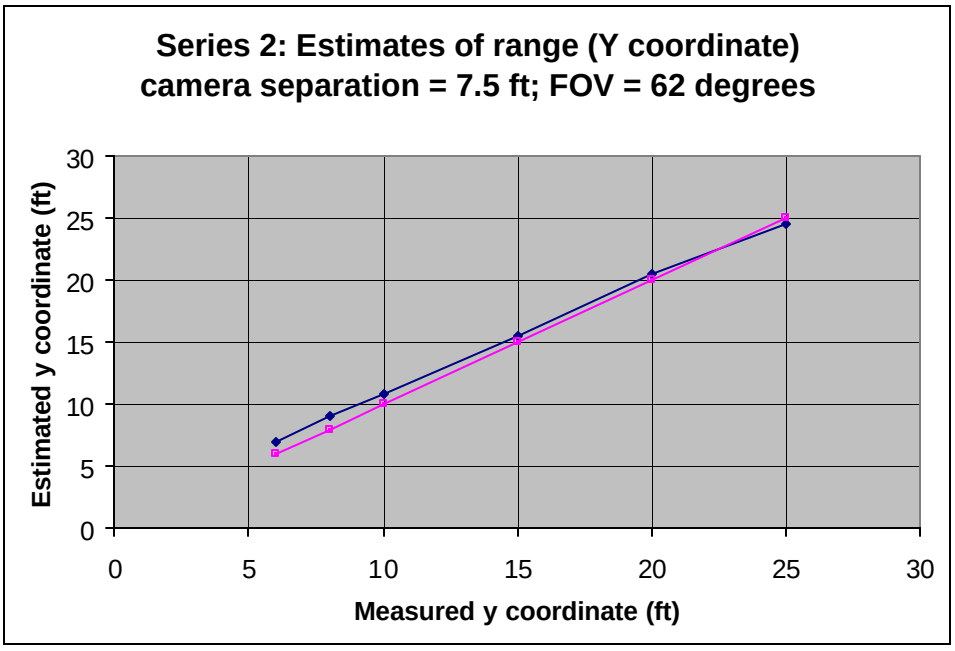


Figure 8. Data series 2 of the y coordinate. Again, the ideal (squares) and experimental (triangles) curves are plotted.

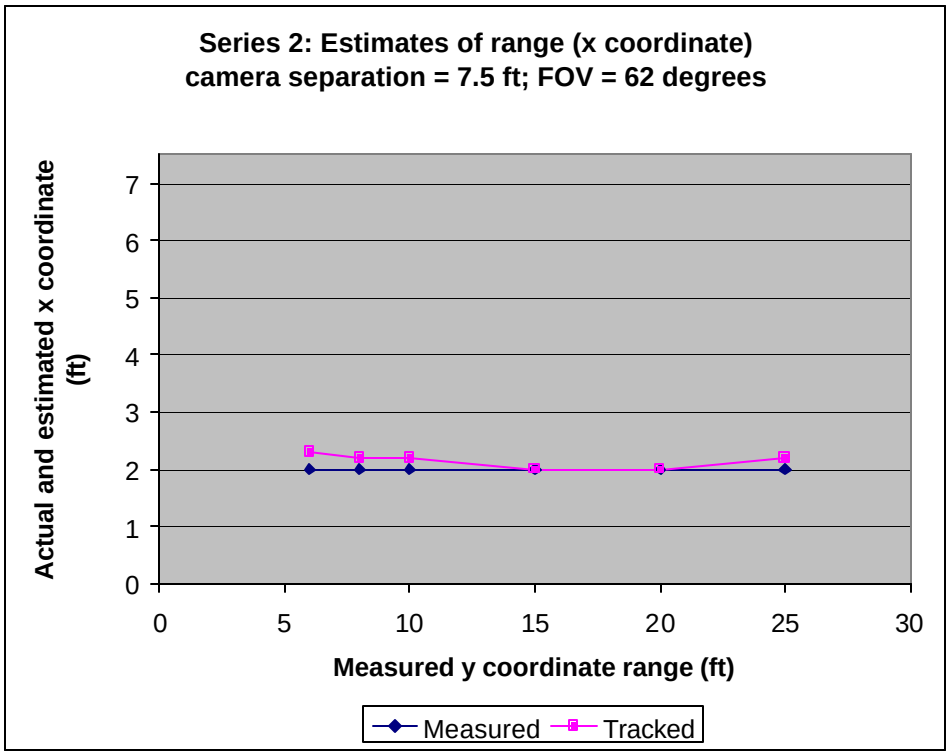


Figure 9. Data series 2 of the x coordinate. Note that this series was done closer to one of the cameras than the other (at 2ft rather than 3.5ft).

Summary

We determined that the following improvements to our first system test will enhance our ability to conduct future tests and further develop the network:

- On-site infrared camera calibration using Sony Glasstrons
- A more consistent source for the infrared tests (not one that is wearing a hood in the snowy weather)
- A marked field for greater accuracy verification in post-processing
- The addition of more modules for more accurate tracking and multiple object tracking
- More real-time readings of tracking status
- Enhanced object recognition routines to assist with real-time tracking
- The use of an "instant" video player, using captured data from the modules, that is being developed by the group

With the information presented above, we feel that we were successfully able to construct a wireless network of prototype sensor modules, and demonstrated this network's ability to track a single object. With the lessons of this test, we will move forward from two modules and one object to tests of three or four modules and multiple objects. We are also working on methods and algorithms from which we can extract more information about the objects within the sensor space. Finally, we are laying the groundwork from which future generations of modules can be constructed that are lighter, faster, and consume less power.